



Vaasan yliopisto
UNIVERSITY OF VAASA

Nikke Tapanainen

Visualisoinnin automaatiojärjestelmän suunnittelu

Tekniikan ja innovaatiojohtamisen akateeminen yksikkö
Tietojärjestelmätieteen pro gradu -tutkielma
Tietojärjestelmätieteen maisteriohjelma

Vaasa 2021

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen akateeminen yksikkö**

Tekijä:	Nikke Tapanainen
Tutkielman nimi:	Visualisoinnin automaatiojärjestelmän suunnittelu
Tutkinto:	Kauppätieteiden maisteri
Oppiaine:	Tietojärjestelmätieteen maisteriohjelma
Työn ohjaaja:	Tuomo Palomaa, Timo Mantere
Valmistumisvuosi:	2021
Sivumäärä:	70

TIIVISTELMÄ:

Visualisoinnin automaatiojärjestelmällä mahdollistettaisiin suurten tietomäärien, erilaisten suunnittelumallien sekä avoimien rajapintojen sisältämän tiedon mallintaminen. Rambollilla ei tällä hetkellä ole käytössä hyvin toimivaa työkalua visualisoinnin automaatioon. Tämän työkalun puute tarkoittaa, että suuria määriä töitä täytyy tehdä käsin ja tämä on aikaa vievää. Visualisoinnin automatisoinnilla voitaisiin suorittaa erilaisia projektiin kuuluvia toimenpiteitä, kuten maastonmuokkausta sekä valaisinpylväiden asettelua. Suunniteltava järjestelmä tarjoaisi siis visualisointiryhmän kautta projektista viimeistellyn 3D-mallin. Tämä malli ei kuitenkaan olisi ainoa käyttötarkoitus järjestelmälle, vaan järjestelmää voitaisiin käyttää myös suunnittelun tukena, tuomalla päivitettyjä 3D-malleja heidän suunnitteluohjelmistoihinsa. Tutkimuksen tavoitteena on siis kerätä tietoa mitä tämänkaltaisen visualisoinnin automaatiojärjestelmän luomiseen tarvittaisiin, mitä asioita olisi hyvä ottaa huomioon sekä mitä todennäköisiä ongelmatilanteita voisi syntyä. Lisäksi tutkitaan muita osa-alueita, jotka liittyvät tällaisen järjestelmän ympäristöön, kuten Rhinocerosin Grasshopperia, Quadri-tietokantasovellusta sekä Unreal Engineä. Grasshopper-sovelluksessa pystytään automatisoinnin avulla tekemään edellä mainittuja valaisinpylväiden asettamisia maastomalliin, mutta Grasshopperin avulla voidaan myös luoda käyttöliittymä koko järjestelmälle. Visualisointiryhmän vetäjä mainitsi, että onnistuneella järjestelmällä voitaisiin vähentää visualisointiryhmän työmäärää. Tutkimusmenetelmänä käytettiin tietojärjestelmätutkimusmallia, ja sen avulla selvitystyötä pilkottiin pienempiin osiin. Itse järjestelmän rakentamiseen käytettiin UML-kieltä ja sieltä eritoten luokka- ja aktiviteettikaaviota. Mahdollisen järjestelmän käytettävyyden tutkintaan otettiin tueksi UX Honeycomb-kaavio ja tämän kaavion joista osa-aluetta tutkittiin järjestelmän kannalta. Toivottuja ominaisuuksia mahdollisella järjestelmällä on muutamia. Erilaisten tietokantayhteyksien muodostuminen täytyisi tapahtua automaattisesti, käyttäjän täytyisi pystyä valitsemaan haluamaansa lisätietoa Maanmittauslaitoksen rajapinnasta sekä järjestelmän käyttöliittymän täytyisi olla käytettävyydeltään hyvä viemättä tietokoneen laskentatehoja. Jos raportissa esitettävä visualisoinnin automaatio järjestelmä tul-taisiin kehittämään, antaisi se arvoa yritykselle sekä työntekijöille, mutta myös medialle sekä mahdollisen projektia koskevan alueen asukkaille. Myös kehityskohteita tämänkaltaiselle mahdollisella järjestelmälle on monia. Järjestelmä voitaisiin siirtää toimimaan verkon välityksellä ja näin pystyttäisiin säästämään Rambollin työntekijöiden aikaa, sillä suurien 3D-mallien luominen on tietokoneen tehoja kuluttavaa, eikä kyseisellä tietokoneella voi tehdä muita toimenpiteitä mallin työstämisen aikana. Toinen mahdollinen kehityskohde olisi lisätä muita rajapintoja, kuin mitä tässä raportissa käydään läpi. Visualisointiin saataisiin lisää uskottavuutta esimerkiksi, jos ympäristö ja ympäröivät rakennukset olisivat saman näköisiä kuin mitä oikeassa maailmassa. Virallisia tuloksia siitä, rakennetaanko tässä raportissa esitettävä järjestelmä ei käydä läpi raportissa, mutta raportti tullaan luovuttamaan henkilöille yrityksessä, joka päättää mahdollisen järjestelmän tuottamisesta.

AVAINSANAT: Grasshopper, Rhinoceros, tietojärjestelmätutkimusmalli, UML, Järjestelmä suunnittelu, visualisointi

Sisällys

1	Johdanto	7
1.1	Tutkimuksen tavoite	8
1.2	Tutkimuksen rakenne	8
1.3	Ramboll	9
2	Järjestelmäsuunnittelun teoria	10
2.1	Tietojärjestelmätutkimus	10
2.1.1	Seitsemän ohjelinjaa tietojärjestelmätutkimukseen	11
2.1.2	Tietojärjestelmätutkimusmalli	12
2.2	Visualisointi	13
2.3	Järjestelmä	13
2.4	UML	14
2.4.1	Luokkakaavio	15
2.4.2	Aktiviteettikaavio	16
2.5	Mahdolliset sovellukset	16
2.5.1	Grasshopper/Rhinoceros	17
2.5.2	Trimble Quadri 2020	18
2.5.3	Unreal Engine	19
2.6	OpenBIM-muodot	19
2.6.1	IFC	20
2.6.2	LandXML	20
2.7	Rajapinta	21
2.8	Käytettävyys ja UX Honeycomb	22
2.9	UI Käyttöliittymä	24
2.10	Tiedonhallinnan periaatteet	25
2.10.1	Tiedon tuottamisen harmonisointi	25
2.10.2	Toimintamallien vakiointi	26
3	Tietojärjestelmätutkimusmalli visualisoinnin automaatio järjestelmälle	27
3.1	Ongelman tunnistus	27
3.2	Ratkaisun tavoitteet	27

3.3	Suunnittelu ja kehitys	28
3.4	Esittely ja arviointi	28
4	Computational Design	29
5	Rhinoceros, Grasshopper ja Quadri	31
5.1	Rhinoceros näkymä ja käyttöliittymä	32
5.2	Grasshopper- näkymä ja käyttöliittymä	34
5.2.1	Grasshopper esimerkki ongelmasta	36
5.2.2	Grasshopper-objektit maanpintaan ja oikeaan suuntaan	38
5.3	Quadri-näkymä	39
6	Järjestelmän mahdolliset rajapinnat ja maanmittauslaitoksen rajapinta	42
7	Suunnitelma järjestelmälle	46
7.1	Haastattelut	47
7.2	UML kuvaukset	47
7.2.1	Luokkakaavio kuvaus	48
7.2.2	Aktiviteettikaavio kuvaus	49
7.3	Käyttöliittymä esimerkki	50
7.4	Hyödyllisiä ominaisuuksia	52
7.5	Haasteet	53
8	Mahdollisen järjestelmän käytettävyyssarvio Honeycomb menetelmällä	56
8.1	Hyödyllisyys	56
8.2	Käyttökelpoisuus	56
8.3	Haluttava	57
8.4	Löydettävä	57
8.5	Saavutettava	57
8.6	Uskottava	58
8.7	Arvokas	58
9	Mahdolliset käyttötapaukset	59
10	Yhteenveto	61
	Lähteet	63

Kuvat

Kuva 1. Tietojärjestelmätutkimusmalli	12
Kuva 2. Luokkakaavio	15
Kuva 3. Rhinocerosin ja Grasshopperin hyväksymiä tiedostomuotoja	17
Kuva 4. Quadrin tarjoamia mahdollisuuksia eri sovelluksiin	18
Kuva 5. UX Honeycomb-kaavio ensimmäinen versio	22
Kuva 6. UX Honeycomb-kaavio toinen versio	24
Kuva 7. Järjestelmän tarvittavat tietovirrat	31
Kuva 8. Rhinoceros-sovelluksen perusnäky	33
Kuva 9. Rhinoceros-sovelluksen vasemman puolen toiminnot	33
Kuva 10. Rhinoceros-sovelluksen yläpalkin toiminnot	34
Kuva 11. Grasshopper Vector-ryhmän komponentteja	34
Kuva 12. Kanvaasin kautta komponentin lisäys	35
Kuva 13. Paneelien käyttö sisään- ja ulostulona	35
Kuva 14. Quadriassa näkymä toiminnallinen ongelma	36
Kuva 15. Ongelma	37
Kuva 16. Virhe automaatio	37
Kuva 17. Onnistunut versio	37
Kuva 18. Lamppujen epäonnistunut sijoittaminen	38
Kuva 19. Lamppujen sijaintien onnistunut automatisointi	39
Kuva 20. Onnistunut automatisointi	39
Kuva 21. Quadri selain ikkunan näkymä	40
Kuva 22. Quadriassa oleva tie	40
Kuva 23. Quadrin valitun objektin ominaisuudet ikkuna	41
Kuva 24. Maanmittauslaitoksen tarjoilu	42
Kuva 25. Maanmittauslaitoksen tietojen metatietoja	43
Kuva 26. Maanmittauslaitoksen tietojen metatietoja	44
Kuva 27. Maanmittauslaitoksen tietojen selitys	44
Kuva 28. Yksinkertainen kuva järjestelmästä	46

Kuva 29. Järjestelmän yksinkertainen esimerkki luokkakaavio	48
Kuva 30. Luokkakaavion "käyttäjä" luokan esittely	49
Kuva 31. UML-aktiviteettikaavio järjestelmästä	50
Kuva 32. Laadittu UI-käyttöliittymä pienempänä	51
Kuva 33. Laadittu UI-käyttöliittymä kokonaan	52
Kuva 34. Quadrin sisältämä tie poikkileikkaus	54
Kuva 35. Quadri ongelmakohta päällekkäisyyksissä	55
Kuva 36. Mahdollisia mallinnuspisteitä	59
Kuva 37. Järjestelmän kautta saatavia tietoja takaisin suunnitteluun	60

1 Johdanto

Laajojen projektien parissa tietoa saattaa muodostua usein monesta eri lähteestä. Tarvittavaa tietoa syntyy paikkatietolähteistä, suunnittelun eri tekniikka-aloilta, operaattoreiden tietolähteistä ja dokumenteista. Eri toimialat yrityksen sisällä sekä sidosryhmissä käyttävät itselleen parhaiten soveltuvia sovelluksia tiedon luomiseen. Ongelmaksi muodostuu usein tilanne, jossa eri sovellukset tai sovelluksien tuotokset eivät välitä tietoja täydellisesti toistensa kanssa. Työntekijöille kertyy valtavasti manuaalista työtä tietojen ylläpidossa sellaisissa tilanteissa, joissa sovellukset eivät välitä tietoa toisilleen. Myös jokin suunnitelmapäivitys tulee päivittää käsin tietokantaan tai siirtotiedostoihin. Tämä vie aikaa ja lisää mahdollisten virheiden syntymistä. Manuaalisen työn lisääntyminen ja päivitykset vaikuttavat myös projektin aikatauluihin, sillä niiden todelliseen määrään on vaikea varautua etukäteen suurissa hankkeissa. Tällaisiin tilanteisiin voidaan löytää ratkaisu automatisoinnin kautta, esimerkiksi rakentamalla järjestelmä, joka yhdistäisi tietoja ja vähentäisi manuaalista työtä.

Alfamen (2016) mukaan yritysten digitalisaatio on lisännyt automatisointia sekä vähentänyt manuaalista työtä. Kun koneille annetaan tehtäväksi rutiininomaiset työtehtävät, työntekijöille jää enemmän aikaa tuottavimpiin työtehtäviin. Automatisointi lisää myös työntekijöiden viihtyvyyttä, sillä puuduttavat rutiinitehtävät vaihtuvat mielekkäämpiin ja haastavampiin tehtäviin. Hevner, March & Park (2004) kirjoittavat artikkelissaan, että tietojärjestelmiä implementoidaan jokaiseen organisaatioon ja tämän tarkoituksena on parantaa organisaatioiden vaikuttavuutta ja tehokkuutta.

Ma (2007) kirjoittaa tieteellisessä artikkelissaan, että visualisointi on muuttunut viimeisen 20 vuoden aikana esitystyökalusta työkaluun, jolla voidaan tehdä löytöjä. Hänen mukaansa ihmisillä on kyky integroida ja luoda uusia vuorovaikutuksen metaforia sekä renderointialgoritmeja. Metaforat ja algoritmit mahdollistavat tiedon interaktiivisen visualisoinnin ja tämä on avain menestykseen.

1.1 Tutkimuksen tavoite

Tutkimuksen tavoitteena on selvittää tiedonhallinnan automatisointia, harmonisoida toimintamalleja suunnitteluprosessin aikana ja tutkia kuinka tietojärjestelmätutkimusmallia voitaisiin käyttää järjestelmän suunnittelussa. Tutkimuksessa kehitetään mahdollisesta järjestelmästä UML-malleja ja näiden kautta tutkitaan järjestelmän tarjoamia mahdollisuuksia. Järjestelmän tulisi noutaa tiettyyn projektiin kuuluvia lähtötietoja ja sekä jalostaa informaatio mahdollisimman käyttökelpoiseksi. Tätä paketoitua mallia voitaisiin sitten jatkotyöstää pelimoottorissa ja viimeistellä renderöitävä malli. Järjestelmän suunnittelussa pyritään siihen, että valmis järjestelmä noutaisi mahdollisimman paljon tietoa suunnittelun palvelimelta sekä avoimien rajapintojen kautta. Pilvipalvelut mahdollistavat sen, että tietoja pystytään muokkaamaan ja että muutokset tulevat näkyville heti itse lopputuloksessa.

Automatisoitu järjestelmä vähentäisi käsintehtyä työtä, ja näin työtunteja voitaisiin käyttää yrityksessä tehokkaammin. Kyseisellä järjestelmällä automatisoidun projektin visualisointi lisäisi puolestaan projektin käytettävyyttä sekä näkyvyyttä, sillä sen avulla suunnitteilla olevaa projektia voitaisiin helpommin esitellä erilaisille sidosryhmille kuten asukkaille, päättäjille ja suunnitteluryhmille.

1.2 Tutkimuksen rakenne

Tässä tutkimuksessa käydään aluksi läpi teoriaa tietojärjestelmätutkimuksesta, tietojärjestelmätutkimusmallisista, sekä mahdollisista sovelluksista, joita voitaisiin käyttää osana uutta järjestelmää. Teoriaosuudessa avataan myös mitä ovat rajapinta, visualisointi, käyttöliittymä, sekä mitä on käytettävyyys UX Honeycomb-kaavion muodossa.

Kun teoriaosuus on ohi, sovelletaan tietojärjestelmätutkimusmallia tutkimuksen kohteeseen, eli visualisoinnin automaation suunnitteluun. Tämän jälkeen tarkastellaan mahdollisen järjestelmän eri sovelluksia, sovellusten näkymiä, sekä niiden ominaisuuksia,

erilaisia tilanteita, joissa mahdollisesta järjestelmästä olisi hyötyä, sekä mitä mahdollisia heikkouksia järjestelmässä voisi ilmetä.

Seuraavaksi avataan itse järjestelmää UML-kuvauksien avulla, sekä pohditaan järjestelmän mahdollista käyttöliittymää, järjestelmän ominaisuuksia ja mahdollisia haasteita mitä järjestelmällä voisi tulla vastaan. Viimeisenä sovelletaan UX Honeycomb-kaaviota suunniteltavan järjestelmän kannalta, ja mietitään Honeycomb kaavion jokaisen osa-alueen tilannetta järjestelmän näkökulmasta.

Mahdolliset käyttötarkoitukset järjestelmälle sekä yhteenveto tutkimuksesta on jätetty tutkimuksen viimeisille sivuille. Käyttötarkoitukset tarkoittavat tässä tilanteessa niitä projektin hankevaiheiden kohtia, joista voitaisiin tehdä 3D-malli projektin aikana. Tällaisia hetkiä ovat ainakin lähtötietomalli ja täysin valmis malli.

1.3 Ramboll

Ramboll (2021) on kansainvälinen konsultointi- ja suunnittelualan yritys, jossa työskentelee 16500 eri alojen ammattilaista. Suomessa Ramboll työllistää 2500 asiantuntijaa. Rambollin palveluja ovat mm. innovatiiviset ratkaisut kaupunkien, liikenteen ja rakennusten suunnittelussa, rakentamisessa sekä ylläpidossa.

Ramboll (2021) panostaa rohkeaan ja avoimeen kehittämiseen mm. Leijonan Luola tyyppisen konseptin kautta. Näin erilaisia ideoita kehitetään ketteriä menetelmiä hyödyntäen yhteistyössä asiakkaiden kanssa. Leijonan luola konseptissa erilaisia ideoita esitetään ammattilaisille, ja samalla tutkitaan voisiko ideoissa olla potentiaalia.

2 Järjestelmäsuunnittelun teoria

Tämän tutkimuksen teoria muodostuu erilaisista osa-alueista, joita ovat muun muassa erilaiset sovellukset, kuten Trimble Quadri, Rhinocerosin Grasshopper ja Unreal Engine. Järjestelmän suunnitteluun liittyviä asioita, kuten UML-kieltä, rajapintaa, visualisointia, käytettävyyttä ja käyttöliittymää avataan myös tässä kappaleessa. Kappale alkaa kuitenkin teorialla järjestelmän suunnittelusta, DSR:stä (Design science research) ja tietojärjestelmätutkimusmallista.

2.1 Tietojärjestelmätutkimus

Tietojärjestelmätutkimus (Design science research) on Reubensin (2016) mukaan kohtalaisen uusi suunnittelulähtöinen tutkimusala. Hänen mukaansa siinä kehitetään tai rakennetaan jokin uusi artefakti ongelman ratkaisuun tai nykyisen järjestelmän parantamiseen. Tässä tutkimuksessa ei virallista järjestelmää tai artefaktia rakenneta, vaan tutkimus keskittyy järjestelmän suunnitteluun ja osa-alueisiin.

Hevner, March & Park (2004) kertovat että, tietojärjestelmätutkimus on prosessi, joka sisältää useamman vaativan toimenpiteen, ja jonka lopputuloksena valmistuu innovatiivinen tuote, eli artefakti. Kun tuote on valmistettu, sitä arvioidaan ja samalla kerätään tärkeää palautetietoa, joka puolestaan tarjoaa parempaa ymmärrystä artefaktista sekä alkuperäisestä ongelmasta. Hevner on yksi niistä tutkijoista, joka sanoi, että tietojärjestelmätutkimuksen päätavoitteena on saavuttaa ymmärrystä ja tietoa ongelman ytimestä rakentamalla artefakti, joka on ratkaisu ongelmaan.

Hevner & Gregor (2013) kertovat, että Wilsonin (2002) työstä ilmeni seuraavanlainen asia. On olemassa kolme tärkeää kysymystä, joita kysytään mahdollisesta tutkimuksesta, kysymykset ovat ”Onko se uusi?”, ”Onko se totta?” sekä ”Onko se kiinnostava?”. Näistä tärkeimpänä on viimeinen kysymys eli, ”Onko se kiinnostava”, jos vastaus tähän on negatiivinen, Hevnerin (2013) mukaan lukijalla ei ole kiinnostusta kahteen muuhun

kysymykseen. Oman uskomuksen mukaan, tähän voitaisiin vielä lisätä kysymys, ”Pystytäänkö sillä tuottamaan voittoa?” ja tämä kysymys tulee mukaan tutkimukseen myös käytettävyyden näkökulmassa.

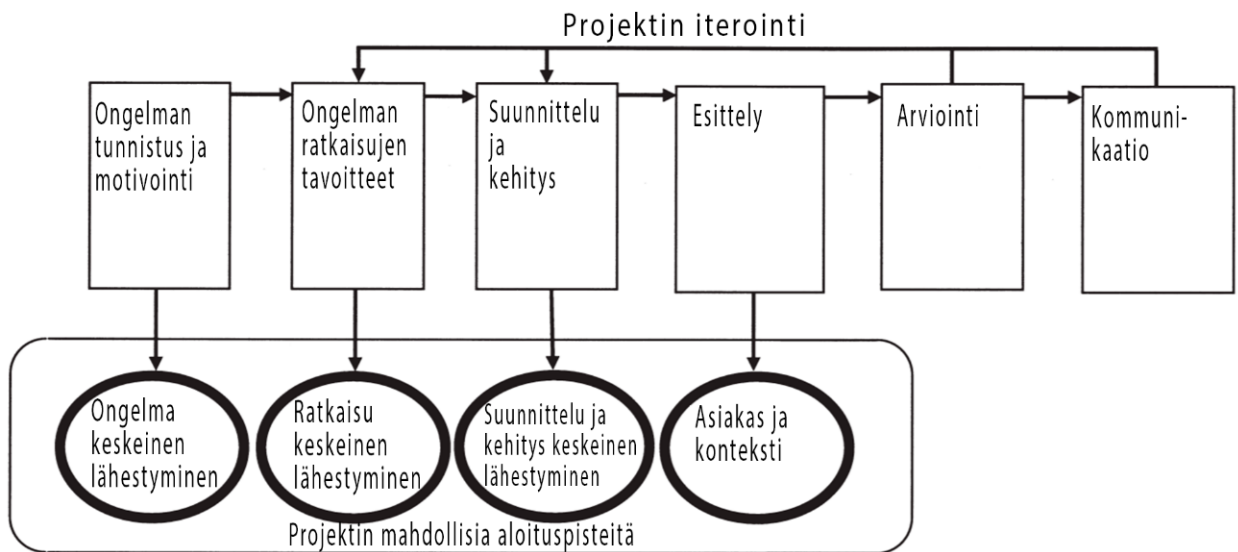
Hevner, March & Park (2004) esittävät artikkelissaan, että tietojärjestelmätutkimuksen peruseriaate on se, että tietoa ja käsitystä tutkimuksen ongelmasta ja sen ratkaisusta hankitaan samalla kun artefaktia rakennetaan ja kun sitä sovelletaan tutkimuksen kohteena olevaan ongelmaan.

2.1.1 Seitsemän ohjelinjaa tietojärjestelmätutkimukseen

Hevner, March & Park (2004) esittävät myös seitsemän ns. ohjelinjaa tietojärjestelmätutkimukseen, joita käyttämällä tutkija löytää parhaan ratkaisun. Ohjelinjoja ovat muun muassa selvät tavoitteet, eli mikä ongelma halutaan ratkaista, miten se voitaisiin ratkaista ja eritoten miksi kyseinen ongelma pitäisi ratkaista. Hevnerin & muiden (2004) mukaan kun tutkija on motivoitunut, hän pystyy antamaan täyden panostuksensa tutkimukselle. Neljäntenä ohjeena Hevner & muut (2004) kertovat kokoontumisen muiden, mieluiten alan ammattilaisten kanssa, sillä kun tutkimuksen päätavoite ja mahdolliset sivutavoitteet ovat selvillä, voidaan muilta ammattilaisilta saada erilaisia ideoita mahdollisiin ratkaisuihin. Ratkaisuja voidaan kerätä monella eri tapaa, mutta tärkeintä on, että sitä ei tehdä yksin. Viidentenä ohjeena Hevner & muut (2004), sanovat ongelman ratkaisujen kehityksen, sekä niiden rajauksen sen mukaan mitkä niistä voisivat oikeasti toimia. Tämän jälkeen ratkaisuja testataan ja viimeisenä vaiheena vertaillaan ratkaisua teorioihin, yleistetään ratkaisun tulos ja viestitään ratkaisusta muiden alan ammattilaisten kanssa, jotta voidaan taas kehittää uusia ratkaisuja sekä artefakteja.

2.1.2 Tietojärjestelmätutkimusmalli

Tutkimuksesta voidaan luoda tietojärjestelmätutkimusmalli eli DSR-malli (Pefers & muut, 2006), joka sisältää kuusi eri vaihetta kuten kuvassa 1. Vaiheet ovat ongelman tunnistus, ongelman ratkaisun tavoitteet, suunnittelu ja kehitys, esittely, arviointi sekä kommunikaatio. Pefersin & muiden mukaan ei ole välttämätöntä, että tutkimus tehtäisiin tässä järjestyksessä, vaan tutkija voi aloittaa työnsä jostain neljästä ensimmäisestä vaiheesta.



Kuva 1. Tietojärjestelmätutkimusmalli

Pefers & muut (2006) kertovat vaiheista enemmän: ensimmäisessä vaiheessa ongelmaa avataan ja siinä kerrotaan ratkaisun ”arvo”. Seuraavaksi kerrotaan ratkaisusta sekä siitä, mitä ratkaisun pitäisi kehittää. Kolmantena vaiheena on itse artefaktin luominen, artefakti voi siis olla esimerkiksi jokin sovellus tai järjestelmämalli. Tässä vaiheessa määritetään myös mitä haluttu artefakti tekee.

Luomisen jälkeen esitetään, miten artefaktilla on saatu ratkaistua ongelma tai osia siitä. Tämän jälkeen tutkitaan, miten hyvin artefaktin luoma ratkaisu on ratkaissut aluksi esitetyn ongelman. Viimeisenä kerrotaan ratkaisusta, artefaktista sekä koko tutkimuksesta muille alan ammattilaisilla ja näin mahdollistetaan taas uusien ratkaisujen kehitys.

2.2 Visualisointi

Xamkin (2021) datavisualisointioppaassa kuvataan visualisointia datan jäsentämisenä ja analysointina sekä viestintänä ja esittämisenä. Tämä data voi olla havaintoja, asiakirjoja tai esimerkiksi mittauksia. Heidän mukaansa visualisoinnin ideana on muokata tätä dataa niin, että se muuttuu ihmisen toiminnan kautta ymmärretyksi ja tietämykseksi. Xamk myös kirjoittaa verkkosivuillaan *”Mikäli datan visualisointi tehdään sijainti huomioon ottaen, voi se auttaa jäsentämään ja ymmärtämään dataa paremmin”*. Tämä huomio on mahdollisen suunniteltavan järjestelmän kannalta merkittävä asia, sillä visualisointia tehtäisiin myös miettien alueen asukkaita.

Xamkin (2021) mukaan visualisointi on siis datan muokkaamista ja esittämistä visuaalisessa muodossa, ilman että tämän tiedon merkitys muuttuu. Haasteeksi Xamk ilmaisee tiedon ymmärtämisen, eli mikä on oleellista ja sen säilyttämisen muuttumattomana. Tästä voi muodostua siis yksi mahdollisen järjestelmänkin haasteista.

DeBoisin (2020) mukaan hyvän visualisoinnin pitäisi täyttää kaksi näkökulmaa visualisoitavasta tiedosta. Visualisoinnin täytyisi näyttää tiedoissa yhteydet visualisointiin, jotka olisivat liian työläitä selittää sanoilla. Toiseksi näkökulmaksi DeBois kertoo, että hyvän visualisoinnin tulisi saada yleisö ymmärtämään nopeasti esitetyt tiedot ja ottamaan huomioon lopputulos.

2.3 Järjestelmä

Martinin (2003) mukaan järjestelmä on suunnittelutaso, joka on yhden tason ylempänä nykyisestä työskentelystäsi. Hoogenraadin (2018) mukaan järjestelmä on organisaatio ongelmiin integroitu ratkaisu, ja tämän selityksen perusteella ohjelmistoa, sovellusta tai työkalua ei voida kutsua järjestelmäksi. Järjestelmä on siis osista, kuten moduuleista ja osajärjestelmistä muodostettu yhtenäinen kokonaisuus.

Ekpo & George (2010) painottavat työssään, että järjestelmätason ymmärrys, analysointi ja suunnittelu ovat tärkeässä osassa, kun varmistetaan monimutkaisten järjestelmien luotettavuutta ja toimivuutta.

2.4 UML

Järjestelmän suunnittelun tueksi voidaan ottaa UML (Unified Modeling Language) kieli, joka sisältää erilaisia kaavioita. Nämä kaaviot ovat Visual Paradigmin (2020) mukaan hyvä apu suunnittelijoille, kun kehitetään uutta sovellusta tai järjestelmää. Visual Paradigm korostaa, että UML-kaavioilla pystytään tarkentamaan, dokumentoimaan sekä rakentamaan artefakteja, liiketoimintamalleja ja muita järjestelmiä, jotka eivät ole sovelluspohjaisia. GeeksforGeeks (2019) kuvailee UML-kieltä seuraavanlaisesti; se ei ole ohjelmointikieli, vaan pikemminkin visuaalinen kieli. Heidän mukaansa UML-kielen päätarkoitus on standardisoida visuaalinen tapa erilaisten järjestelmien suunnittelun esittämiseen.

Lon & Huangin (2010) mukaan UML-kielestä on tullut standardi niin kutsuttuun olio-ohjelmistokehitykseen. Olio-ohjelmointia voidaan siis käyttää monenlaisten järjestelmien kuvaukseen. Tutkijoiden mukaan UML-kielessä yhdistyvät tärkeät ohjelmistotekniikat, jotka myöhemmin auttavat ohjelmoijaa määrittämään, ylläpitämään ja kehittämään ohjelmistojärjestelmiä. UML-kieltä on käytetty menestyksekkäästi myös ohjelmistoteollisuudessa monimutkaisten järjestelmien mallintamiseen.

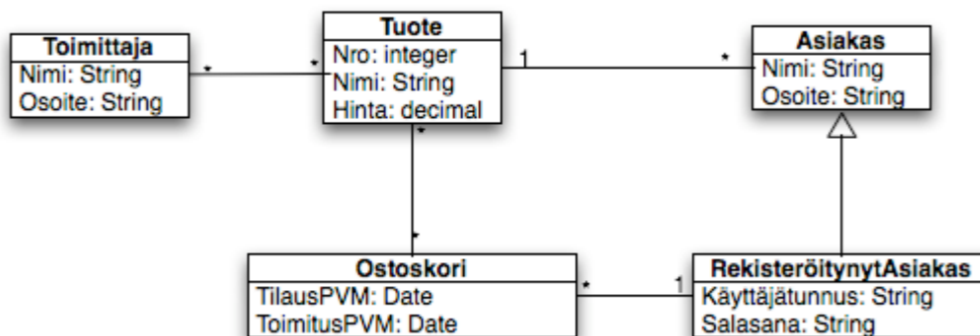
Tämän järjestelmän suunnittelun avuksi valittiin UML-kielestä kaksi eri kaaviota. Luokka-kaavio sekä tilakaavio. Syyksi tähän todettiin, että luokka- ja aktiviteettikaavio tarjoavat parhaan avun suunnitella olevalle mahdolliselle järjestelmälle.

2.4.1 Luokkakaavio

UML luokkakaavio on Visual Paradigmin (2020) mukaan graafinen tapa objektisuuntautuneiden järjestelmien rakentamiseen ja visualisointiin. Luokkakaavio on eräänlainen staattinen rakennekaavio, joka kuvaa järjestelmän rakennetta näyttämällä järjestelmän luokat, attribuutit, metodit ja suhteet muihin objekteihin kuten kuvassa 2.

Tutorialspoint (2021) määrittää luokkakaavion tarkoitukseksi seuraavat neljä eri asiaa. Sovelluksen staattisen näkymän analysoinnin ja suunnittelun, järjestelmän vastuiden kuvaamisen, perustana olemisen komponentti- ja käyttöönottokaavioille sekä viimeisenä mahdollisuuden eteenpäin ja taaksepäin suunnittelulle.

Luokkakaavioesimerkki: Ostoskori



Kuva 2. Luokkakaavio

Jaiwain (2017) tieteellisessä artikkelissa mainitaan, että luokkakaaviot edustavat ohjelmiston sisältämiä kokonaisuuksia. Nämä kokonaisuudet koostuvat luokista, toiminnoista attribuuteista ja luokkien välisistä suhteista. Jaiwain mukaan luokkakaavioiden tekoon vaaditaan ymmärrystä myös siitä, mihin kyseistä järjestelmää suunnitellaan. Esimerkiksi pankkiin suunniteltavan järjestelmän kehittäjällä täytyy olla kokemusta finanssialasta. Taas jos suunnitelmalla on liikaa vaatimuksia, voi luokkakaaviosta tulla sekava, sinne voi syntyä virheitä ja sen kehittäminen voi viedä liikaa aikaa.

2.4.2 Aktiviteettikaavio

Lucidchartin (2021) mukaan aktiviteettikaavio on oikeastaan vuokaavio, joka näyttää järjestelmän suorittamat toiminnot. Heidän mukaansa aktiviteettikaavio on hieman kuin käyttäytymiskaavio, sillä aktiviteettikaavio kertoo mitä täytyy tapahtua suunniteltavissa olevassa järjestelmässä. Aktiviteettikaavion parhaita ominaisuuksia on sen kyky kuvata eri ”askeleet” eli tilat mitkä tapahtuvat järjestelmässä.

Aktiviteettikaavion piirtoon SmartDraw (2021) kertoo selkeät ohjeet. Eri toiminnoilla: esimerkiksi aloituksella, tapahtumalla, kysymyksillä ja lopetuksella on omat piirrosmerkkinsä. Selkeästi ja loogisesti vasemmalta oikealla kuvattu aktiviteettikaavio on helposti lähestyttävä ja siitä lukija saa hyvän kuvan järjestelmän eri aktiviteeteistä ja järjestelmän risteyskohdista.

2.5 Mahdolliset sovellukset

Mahdollisen järjestelmän kehittämiseen käytettäviä sovelluksia lähdetään etsimään aluksi Rambollin jo käyttämistä sovelluksista. Trimblen valmistama Quadri-sovellus tarjoaa käyttäjilleen tietopankin ja onkin jo tällä hetkellä Rambollilla käytössä. Tätä tietoa pitäisi vain jotenkin pystyä käsittelemään. Eri alojen suunnittelijoiden sekä automatisoinnin kehittäjien suuressa suosiossa on Rhinocerosin lisäosa Grasshopper. Grasshopperin yksi puute on kauniin visualisoinnin luominen, jonka taas erilaiset pelimoottorit suorittavat paljon paremmin. Rambollissa toimii Epic Gamesin Unreal Engine pelimoottorilla töitä tekevä visualisointiryhmä.

2.5.1 Grasshopper/Rhinoceros

3dNativesin (2020) mukaan Rhinoceros, kutsumanimeltään Rhino, on yksi suosituimmista 3D-mallinnusohjelmistoista. Rhinoa käyttävät niin arkkitehdit, insinöörit, kuin korusepät ja sitä kehitetään jatkuvasti lisää. Rhino on CAD-sovellus (Computer aided design), jolla käyttäjä pystyy rakentamaan erilaisia muotoja piirustuksesta, luonnoksesta tai esimerkiksi 3D skannauksesta (3dNatives, 2020). Sovelluksen hienouksia on myös sen käytettävyyys monien muiden suunnittelu-, prototyöskentely- ja renderöinti sovellusten kanssa sekä sen käyttö muiden tiedostomuotojen kanssa kuten näemme kuvassa 3.

Yksi esimerkki Rhinon laajoista käyttömahdollisuuksista ilmenee tutkijoiden Wang, Yang ja Ren (2011) tutkimuksessa. He loivat Rhinoon pohjautuvan järjestelmän ihmismallien tekemiseen käyttäen Rhinon käyräobjekteja. Heidän mielestään Rhino on ammattilais- tuotesuunnittelualusta ja verrattuna muihin ihmismallinnusohjelmistoihin, Rhino on yhteensopivampi. Rhino voidaan myös heidän mukaansa helposti yhdistää suunnitteluympäristöön, jossa tutkitaan ihmisen ja koneen vuorovaikutusta.

**.3dm .3ds .sat .ai .amf .dwg .dxf .cd .iges
.sat .dgn .m .pdf .ply .sldprt .x .pts .svg .dae
.skp .stl .step .stp .obj .rib .xgl .x_t .zpr .fbx**

Kuva 3. Rhinocerosin ja Grasshopperin hyväksymiä tiedostomuotoja

Rhinocerosin (2020) mukaan sovelluksella käyttäjä pystyy luomaan, muokkaamaan, dokumentoimaan, analysoimaan ja animoimaan erilaisia NURB-käyriä (Non-Uniform Rational B-Splines), tasoja, kiinteitä geometrisiä muotoja, pistepilviä ja monikulmiomalleja eli meshejä.

Grasshopperin yhtä mahdollisuutta tutkivat Hsu & muut (2020), kun he rakensivat järjestelmän, jolla pystytään esittämään ja muokkaamaan arkkitehtonisia malleja virtuaalitodellisuus esityksen aikana. Koska järjestelmä toimii Grasshopperin ja Rhinon päällä,

jatkuva tiedonmuokkaus oli mahdollista ja muutokset näkyivät myös virtuaalimaailmassa välittömästi. Heidän tutkimuksessansa ilmeni myös että, Rhino ja Grasshopper olivat jo valmiiksi arkkitehtien suosiossa.

2.5.2 Trimble Quadri 2020

Trimblen (2021) Quadri-sovellus on avoin yhteinen dataympäristö mallien jatkuvaan jakamiseen samalla kun malleja tuotetaan. Käyttäjä jakaa tietomallivaatimusten mukaisesti projektin ydinmallin, jota Quadri tarjoaa muille projektin osallisille. Malli on aina ajan tasalla ja sitä on helppo jakaa kaikille osapuolille.

Quadria voitaisiin ehkä käyttää järjestelmän tiedonlähteenä, sillä Trimblen Quadrin (2021) rajapinnan tuki mahdollistaa helpon tietohaun sen pilvipalvelusta. Kun tietoja käytetään rajapinnan kautta, vähentyy turha tiedostojen kopioiminen. Quadrilla on erilaisia liittimiä eri sovelluksiin, kuten Tekla Structuressiin, Grasshopperiin sekä Revittiin (Kuva 4.), joista jokaista käytetään Rambollissa.



Kuva 4. Quadrin tarjoamia mahdollisuuksia eri sovelluksiin

2.5.3 Unreal Engine

Epic Gamesin (2021) valmistama Unreal Engine, on monesta erilaisesta kehitystyökalusta koottu paketti kenelle tahansa, joka työskentelee reaaliaikaisen teknologian kanssa. Unreal Enginellä pystytään rakentamaan suunnitelmiin perustuvaa visualisointia sekä elokuvallisia kokemuksia hyvällä laadulla. Rambollin visualisointiryhmä toimii suurimmaksi osaksi käyttäen Unreal Engineä, luodessaan erilaisiin suunnittelu materiaaleihin perustuvia viimeisteltyjä 3D-malleja.

Li, Li & Liu (2013) toteuttivat Unreal Enginellä virtuaalisen kuvauksen kaivosjärjestelmästä, jolla voidaan edistää tuottavuutta, turvallisuutta ja tehokkuutta nykyaikaisissa hiilikaivoksissa. Järjestelmä käytti oikean maailman tietoja ja sitä onkin käytetty Dongtan hiilikaivoksissa. Heidän mukaansa Unreal Engine valittiin, koska sillä pystyttiin mallintamaan hiilikaivosten tekniset prosessit niin maan alla kuin päällä.

2.6 OpenBIM-muodot

BuildingSMART:in (2021) verkkoartikkelissa ilmenee, että OpenBIM laajentaa BIM:in (Building Information Modeling) etuja, parantamalla digitaalisen datan käytettävyyttä, saatavuutta ja hallintaa. OpenBIM:in ydinidea on yhteistyöprosessi, jolla ei ole omaa myyjää tai omistajaa, eli se on myyjäneutraali.

BuildingSMART:in (2021) mukaan OpenBIM-prosessit ovat jaettavia projektitiedostoja, jotka tukevat saumatonta yhteystyötä kaikille projektiin osallistuville, olivat he sitten suunnittelijoita tai 3D-mallintajia. OpenBIM mainostaa itseään, että se helpottaa yhteen toimivuutta muiden sovellusten kanssa hyödyntäen projekteja ja resursseja koko projektin elinkaaren ajan.

Baldwin (2018) yksinkertaistaa OpenBIM nimityksen. Hänen mukaansa OpenBIM tarkoittaa työskentelyä BIM:in kanssa, mutta avoimilla standardeilla. Avoimet standardit

tarkoittavat Baldwinin mukaan esimerkiksi PDF-tiedostoa Wordin perinteisten DOC-tiedoston tilalla. PDF-tiedoston kanssa työskentely on verrattavissa OpenBIM:iin. Työskentely aloitetaan luomalla malli kaupallisella mallinnusohjelmalla ja sitten malli muutetaan avoimen standardin muotoon, kuten IFC-tiedostomuotoon, näin muut henkilöt saavat mallin auki sekä voivat tarkastella sitä, mutta muutoksia he eivät ole kykeneviä tekemään.

2.6.1 IFC

IFC on FILEFORMAT (2021) sivuston mukaan tiedostomuoto, joka vahvistaa kansainväliset standardit rakennusobjektien sekä niiden ominaisuuksien vientiä ja tuontia varten. IFC-tiedostomuodolla tiedostoa voidaan avata eri ohjelmistosovelluksissa ja tämän tiedostomuodon tavoitteena on parantaa kommunikaatiota, toimitustaikaa, tuottavuutta ja laatua läpi rakennuksen elinkaaren.

Baldwin (2018) kuvailee IFC-tiedoston perinteistä tiedostokulkua seuraavanlaisesti. Suunnittelija luo rakennuksesta mallin ja muuttaa sen IFC-tiedostomuotoon. Muunnetun tiedoston hän antaa LVI-suunnittelijalle, joka yhdistää IFC-tiedoston omaan suunnitteluohjelmaansa. LVI-suunnittelija pystyy tekemään erilaisia kokeita omalla sovelluksellaan IFC-tiedoston perusteella, sillä IFC-tiedosto sisältää riittävästi tietoa mallista, jotta erilaisia simulointeja voidaan tehdä. LVI-suunnittelija ei kuitenkaan pysty leikkaamaan halua- maansa aukkoa saamaansa rakennuksen IFC-malliin, joten hänen täytyy antaa aukon tiedot ensimmäiselle suunnittelijalle ja odottaa rakennuksesta uutta IFC-tiedostoa.

2.6.2 LandXML

AQUAVEO (2019) verkkosivuston mukaan LandXML on tiedostomuoto ilman patenttia. LandXML tallentaa tietoa kuten pisteitä ja tasoja ja tämä helpottaa erilaisten pintojen jakamisen ohjelmien välillä. Monet sovellukset esimerkiksi CAD tukee tietojen tallentamista LandXML muotoon.

Syrjä (2018) kirjoittaa verkkoartikkelissaan, että LandXML:n rakenne on samatyypinen kuin XML-tiedostolla, joten sitä voidaan katsoa normaalilla tekstieditorilla. LandXML-tiedostomuoto on siis hierarkkinen mutta hyvin monimutkainen, eli sen purku manuaalisesti ei ole toimivaa. Yksittäinen LandXML-tiedosto voi Syrjän mukaan sisältää koko työmaan kaikki suunnitelmat. Tarvittavia osioita täytyy siis irrottaa tästä isosta päätiedostosta.

2.7 Rajapinta

Valjaksen (2021) mukaan rajapinta mahdollistaa ohjelmiston yhdistämisen toiseen ohjelmistoon, eli integraation, tietojen siirtämiseksi. Rajapinnan avulla käyttäjä voi noutaa tai tuoda tietoja ohjelmistosta, luomalla erilaisia pyyntöjä ohjelmistolle rajapinnan kautta. Avoin rajapinta (2014) kuvailee avointa rajapintaa rajapinnaksi, jota kuka tahansa voi käyttää ilman rajoittavia ehtoja, ja jonka kaikki ominaisuudet ovat julkisia. Toisin sanottuna kenellä tahansa on oikeus rakentaa järjestelmä, joka käyttää avoimen rajapinnan tietoja, ilman rajapinnan omistajan lupaa. Rajapinnan ja avoimen rajapinnan ero on siis siinä voiko sitä käyttää vapaasti ilman niin sanottua API-avainta. API-avain on Valjaksen (2021) mukaan hieman kuin oman asunnon avain, ilman avainta käyttäjällä ei ole oikeutta rajapinnan tietoihin.

Renin & muiden (2020) artikkelista voimme todeta, että rajapinta voi sisältää oman spesifioidun dokumentin, joka sisältää tärkeää tietoa kyseisestä rajapinnasta, kuten Java-API dokumentti. Tutkijoiden mukaan rajapinnan dokumentti voi sisältää sopimuksia, ohjeita ja rajoituksia, jotka määrittelevät mitä kehittäjät saavat tai eivät saa tehdä rajapinnan avulla.

2.8 Käytettävyys ja UX Honeycomb

Suunnitteilla olevan järjestelmän käytettävyyttä voidaan tutkia esimerkiksi käyttöliittymän suunnittelun kautta soveltamalla UX Honeycomb-menetelmää, mutta tämä arviointi tapahtuu myöhemmin kappaleessa seitsemän. UX Honeycomb on Peter Morvillen (2004) kehittämä kaavio käyttäjäkokemuksen havainnollistamiseen. Honeycomb-kaavio koostuu seitsemästä eri osa-alueesta, joista jokainen kuvaa omaa käytettävyyden palasta. Nämä palaset ovat ”hyödyllinen”, ”käyttökelpoinen”, ”haluttava”, löydettävä”, ”saavutettava”, ”uskottava” ja ”arvokas”.

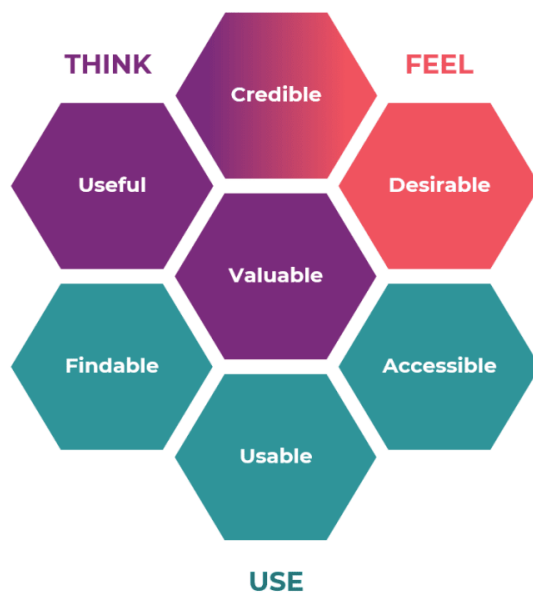
Morvillen (2004) mukaan Honeycomb-kaavio tarjoaa monta käyttötarkoitusta yhtä aikaa. Se toimii hyvin keskustelun edistämisessä. Toisena käyttötarkoituksena se tarjoaa ihmisille tarpeet määritellä prioriteetteja, esimerkiksi onko tärkeämpää, että verkkosivu on haluttava vai käyttökelpoinen tai kenties uskottava. Morvillen mukaan todellisuudessa näihin prioriteetteihin vaikuttavat konteksti, sisältö sekä itse käyttäjät.



Kuva 5. UX Honeycomb-kaavio ensimmäinen versio

Usability verkkosivusto (2021) avaa tarkemmin Morvillen Honeycomb-kaaviota (kuva 5). Hyödyllinen tarkoittaa yksinkertaisesti sitä, että järjestelmän tai sovelluksen täytyisi täyttää tarve mihin se on suunniteltu. Käyttökelpoinen on myös nimensä mukainen, arvioitava järjestelmää täytyisi olla helppo käyttää. Haluttavuus lisääntyy esimerkiksi erilaisilla kuvilla, brändeillä tai muilla muotoiluelementeillä ja näin herätetään tunteita ja arvostusta käyttäjässä. Löydettävyydellä tarkoitetaan, että järjestelmän tai verkkosivun sisällön täytyy olla loogisesti järjestetty, eli että sieltä löytää haluamansa. Saavutettavuus taas tarkoittaa sovelluksen käyttökelpoisuutta myös vammaisten käyttäessä sitä. Viimeisenä on arvokkuus, tämä tarkoittaa, että käyttäjien täytyy uskoa ja luottaa sisältöön.

Macpherson (2019) mainitsee muutamia lisähuomioita mitä ei ilmennyt Usabilityn pohdinnassa käytettävyydestä. Hyödyllisyydestä hän mainitsee, että se on eniten käyttäjän käsissä. Videopelipelaaja voi valita seuraavan pelinsä sen mukaan, onko pelissä mahdollista tehdä hahmon muokkausta, kun taas toiselle tämä voi olla täysin turha aihe. Käytettävyydessä Macphersonin mukaan voidaan laskeutua niin alas, että lasketaan vaadittavat napin painallukset, jotta käyttäjä pääsee haluamaansa lopputulokseen. Saavutettavuudessa Macpherson mainitsee, että helppokäytettävyys on monissa maissa lain määräämä, joten sen puuttuminen on vakava asia. Arvokkuutta Usability ei ollut avannut ollenkaan, mutta Macpherson mainitsee arvokkuudesta seuraavia asioita. Tuotteen täytyy tuottaa arvoa yritykselle ja sen kautta sen asiakkaille, kuitenkin hänen mukaansa on muistettava, että voitto tulee ennen saumatonta käyttökokemusta. Macpherson myös jakaa kaavion osat ryhmiin käyttö, tunto ja ajatus (kuva 6.).



Kuva 6. UX Honeycomb-kaavio toinen versio

2.9 UI Käyttöliittymä

Käyttöliittymän suunnittelussa täytyy ottaa huomioon monia asioita ja kuten Graafinen (2015) kertoo verkkojulkaisussaan ”parhaat käyttöliittymät ovat niitä, joihin et kiinnitä huomiota niitä käyttäessäsi”. Käyttöliittymän suunnittelussa on siis hyvä muistaa muutamia asioita, joita Graafinen on listannut.

Tämän tutkimuksen kannalta Graafisen (2021) tärkeimmät huomiot ovat ”Pysy johdonmukaisena” sekä ”Anna palautetta” ja käyttöliittymän esimerkkiversio onkin tehty eritoten ajatellen näitä kahta neuvoa. Muranen & Harmainen (2021) korostaa käyttöliittymäsuunnittelun tärkeintä asiaa, eli miten tuotteen tai palvelun käyttämisestä tehdään helppoa eri käyttäjille. Käyttöliittymäsuunnittelussa kirjataan ylös käyttäjille tarpeelliset toiminnot sekä käyttöliittymän rakenne ja suunnittelussa korostuu käyttäjien käyttöympäristöjen sekä tarpeiden ymmärtäminen.

Palautteen antaminen on tärkeää tilanteissa, joissa käyttäjälle voi syntyä epätietoinen tila. Myös Graafinen mainitsee, että käyttäjää ei saa ikinä jättää epätietoiseen tilaan, vaan hänen on saatava tietää, onnistuiko käyttäjä siinä mitä hän yritti tehdä. Tällainen palaute lisää hyvää käyttäjäkokemusta ja koko järjestelmän käytettävyyttä. Johdonmukaisuus lisää myös käytettävyyttä omalta osaltaan, mutta myös nopeuttaa järjestelmän käyttöä. Johdonmukaisuutta voi myös lisätä painikkeisiin lisätyillä teksteillä tai erilaisilla muodoilla kuten nuolilla.

2.10 Tiedonhallinnan periaatteet

Tiedonhallinta on Rambollin sisäisen tiedonhallinta ohjeistuksen mukaan tiedon keräämistä, organisointia ja tallentamista. Tärkeää on, että tieto saadaan käyttöön tarkoituksenmukaisesti ja hallitusti. Kun tiedonhallinta on oikein toteutettu, se tuo selkeyttä toimintaan, on siis muistettava, että tiedonhallinta ei ole vain IT-osaston vastuulla.

Röyskö (2021) kirjoittaa verkkoartikkelissaan, että tiedonhallinnan rooli eri organisaatioiden menestykselle kasvaa. Hänen mukaansa tiedon määrä kasvaa jatkuvasti ja sen hallitsemisesta tulee haastavampaa. Yhtenä haasteena Röyskö mainitsee tiedon pirstaloitumisen eri järjestelmiin ja tietopankkeihin. Tähän liittyy myös toinen iso ongelma; tiedon siirtäminen eri kohteiden välillä.

2.10.1 Tiedon tuottamisen harmonisointi

RAKLI (2021) kirjoittaa verkkosivustollaan, että tiedon harmonisointi mahdollistaa tiedon yksikäsitteisen käsittelyn. Yksikäsitteinen käsittely on edellytyksenä tiedon koneluettavuudelle. Heidän mukaan kiinteistö- ja rakentamisalan yksi haaste on tiedonhallinnan standardien määrä sekä päällekkäisyys ja siitä seuraava käsityön määrä erilaisissa tiedonsiirroissa. Longbotham, Kontgis & Maquire (2018) huomasivat omassa tutkimuksessaan,

että tiedon harmonisoinnilla voidaan saavuttaa paljon suurempi kehitysnopeus järjestelmälle ja samalla pystytään luomaan monimutkaisempia järjestelmiä.

2.10.2 Toimintamallien vakiointi

Compeanin (2016) mukaan, toimintamalli on ensimmäinen taso yrityksen arkkitehtuurin toteutuksessa. Käytännössä toimintamallit ovat liiketoimintaprosessien standardointia ja integrointia. Toimintamalli on liiketoimintamallin ensimmäinen ilmentymä, sillä se osoittaa kuinka yrityksen liiketoimintayksiköt luovat, toimittavat ja vangitsevat yrityksen arvon. Yhdysvalloissa ja Euroopassa tehtyjen tutkimusten perusteella, voidaan oman liiketoimintallinsa luoneet yritykset jakaa neljään eri ryhmään.

Compean (2016) kirjoittaa artikkelissaan seuraavanlaisesti ryhmistä. Ensimmäinen ryhmä on niin sanottu koordinoituintoimintamalli. Tälle mallille on ominaista yhteiset asiakas-, toimittaja- ja tuotetiedot, mutta heillä on kuitenkin toiminnallisesti ainutlaatuiset liiketoimintayksiköt. Toisena mallina hän mainitsee yhdistämisen toimintamallin. Se perustuu integroituihin liiketoimintaprosesseihin, joissa asiakkaat ja toimittajat jaetaan maantieteellisesti. Yhdistäminen perustuu ensisijaiseen prosessien ja tietojen joukkoon, joka voidaan määrittää dynaamisesti suoritettavaksi kunkin liiketoiminta yksikön toiminnassa.

Kolmantena mallina hän kertoo hajauttamisen toimintamallin. Tässä mallissa liiketoimintayksiköillä on vain vähän asiakkaita sekä toimittajia. Liiketoimintayksiköiden toiminta on ainutlaatuista ja niiden liiketoimet ovat itsenäisiä, mutta yksiköt kuitenkin hyödyntävät yhteisiä jaettuja palveluita, joita voidaan integroida heidän omiin yksikköihinsä. Viimeinen malli on hieman kuin hajauttaminenkin, mallissa on vähän asiakkaita sekä toimittajia, mutta liiketoimintayksiköt hyödyntävät yhdistettyä lähestymistapaa liiketoimintaprosessien integrointiin sekä standardointiin. Tämän mallin liiketoimintayksiköt ovat siis toimintansa kannalta hyvin samalaisia.

3 Tietojärjestelmätutkimusmalli visualisoinnin automaatio järjestelmälle

Sovittamalla tehtyä tutkimustyötä mahdollisesta visualisoinnin automaatiojärjestelmästä tietojärjestelmätutkimusmalliin, voidaan tutkimustyö pilkkoa osiin. Näitä pilkotuja osia tutkitaan tarkemmin tässä kappaleessa. Kuten mainittu aiemmin, nämä osiot ovat seuraavat; ongelman tunnistus, ongelman ratkaisun tavoitteet, suunnittelu ja kehitys, esittely, arviointi sekä kommunikaatio.

Kuten luvussa kaksi mainittiin, tietojärjestelmätutkimusmallissa on neljä eri vaihtoehtoa siihen mistä tutkimus aloitetaan, eli minkälainen lähestymistapa valitaan tutkimukseen. Tähän työhön valittiin suunnittelu pohjainen lähestymistapa, eli ryhdyttiin tutkimaan, voitaisiinko järjestelmää, joka automaattisesti visualisoisi tietoa, rakentaa Rambollilla käytössä olevista sovelluksista. Työssä suunniteltiin myös minkälaisia ominaisuuksia järjestelmä voisi sisältää ja miten mahdollinen järjestelmä olisi järkevintä kehittää.

3.1 Ongelman tunnistus

Ensimmäisessä palaverissa (T. Palomaa, palaveri 31.8.2020), selvisi tutkimustyömme kohde eli visualisoinnin automaation puute. Yrityksessä ei ole järjestelmää, mistä käyttäjä pystyisi valitsemaan visualisointiin tarvittavat tiedot. Tällaisen järjestelmän puute aiheutti yrityksessä paljon toistoa vaativaa työtä, ja juuri tässä Palomaa havaitsi mahdollisen tutkimustyön. Tutkimustyöksi valittiin siis, että pystyttäisiinkö visualisointia automatisoida, ja eritoten suunnitelma siitä miten se olisi järkevintä tehdä.

3.2 Ratkaisun tavoitteet

Ratkaisuksi tähän ongelmaan valittiin tutkimustyö, jossa lähdettiin tutkimaan mahdollisuuksia rakentaa järjestelmä, joka noutaa ja kerää yhteen erilaiset tietolähteet ja

yhdistää geometriset muodot sekä mallintaa geometriaa informaation perusteella (esim. MML rakennus kerroslukumäärä)

3.3 Suunnittelu ja kehitys

Mahdollisen järjestelmän vaiheita pohdittiin useissa työpajoissa. Nykytilanteiden tunnistaminen ja uuden järjestelmän toimintamallit vaativat prosessien ja toimintamallien dokumentointia sekä niiden kehittämistä.

Järjestelmän kehitys jää odottamaan tutkimuksen vastaanottoa, tutkintaa siitä olisiko siitä todella hyötyä yritykselle sekä varmistusta, että kyseisen järjestelmän tuottaminen olisi todellisuudessa mahdollista.

3.4 Esittely ja arviointi

Järjestelmän suunnitelma tullaan esittämään virallisesti 2021 syksyllä, joten valitettavasti siihen liittyvää arviointia tai kommunikaatiota ei käydä läpi tässä raportissa. Suunnitelman arviointina voidaan hyvin käyttää päätöstä siitä, lähdetäänkö suunniteltua järjestelmää rakentamaan.

4 Computational Design

Lameran (2020) mukaan Computational Design eli laskennallinen suunnittelu, on laskennallistekniikoiden ja suunnittelutekniikoiden yhdistelmä. Laskennallistekniikoiden lisääminen suunnittelun työkulkuun muokkaa sitä tapaa, jolla ihmisten rakentavat rajapintoja, rakennuksia tai palveluja. Lameran mukaan kiinteiden objektien määrittelemisen sijasta suunnittelijoiden on määriteltävä prosessi, jolla objekti luodaan. Luotu prosessi toimii erilaisilla algoritmeilla. Ihmiset eivät luo lopputuloksia käsin, vaan lopputulokset luodaan automaattisesti erilaisten ohjeiden, muuttujien ja parametrien avulla.

Suunnittelijat ovat jo tottuneen työskentelemään erilaisten piirtotyökalujen, kuten Rhinon kanssa, toteuttaakseen omia ideoitaan. Lameran (2020) artikkelissa ilmenee, että nämä piirtotyökalut auttavat suunnittelijoita siirtymään abstraktista käsityksestä konkreettiseen esitykseen. Tällainen siirtymä on käytössä monilla aloilla ja tulos syntyy suunnittelijan aivojen generatiivisen prosessin tuloksena. Erilaisten objektien kuten viivojen ja muotojen piirtämisen sijasta, suunnittelijoiden on määritettävä kaikki laskennalliset ohjeet, muuttujat ja parametrit lopputuloksen saavuttamiseksi.

Kilkelly (2016) määrittää artikkelissaan, että laskennallinen suunnittelu on laaja termi. Se kattaa erilaisia toimintoja, tehtävien automatisoinnista, suunnittelun luomiseen. Selkärangana laskennallisessa suunnittelussa on kuitenkin visuaalisen ohjelmointityökalun käyttö. Kilkelly listaa viisi tapaa hyötyä laskennallisesta suunnittelusta.

Ensimmäisenä tapana hän mainitsee mahdollisuuden rakentaa useita suunnitteluvaihtoehtoja. Luomalla koodi suunnittelusäännöistä laskennallisessa kehyksessä, voidaan luoda satoja erilaisia malleja näiden suunnittelusääntöjen pohjalta.

Toiseksi tavaksi Kilkelly (2016) mainitsee, että erilaisilla laskennallisensuunnittelun-sovelluksilla voidaan helposti päästä käsiksi mallien sisältämään tietoon. Kilkelly mainitsee erikseen, että Revit sovelluksella voidaan viedä kaikki mallin sisältämä tieto Exceliin, muokata sitä Excelissä ja tuoda takaisin Revittiin.

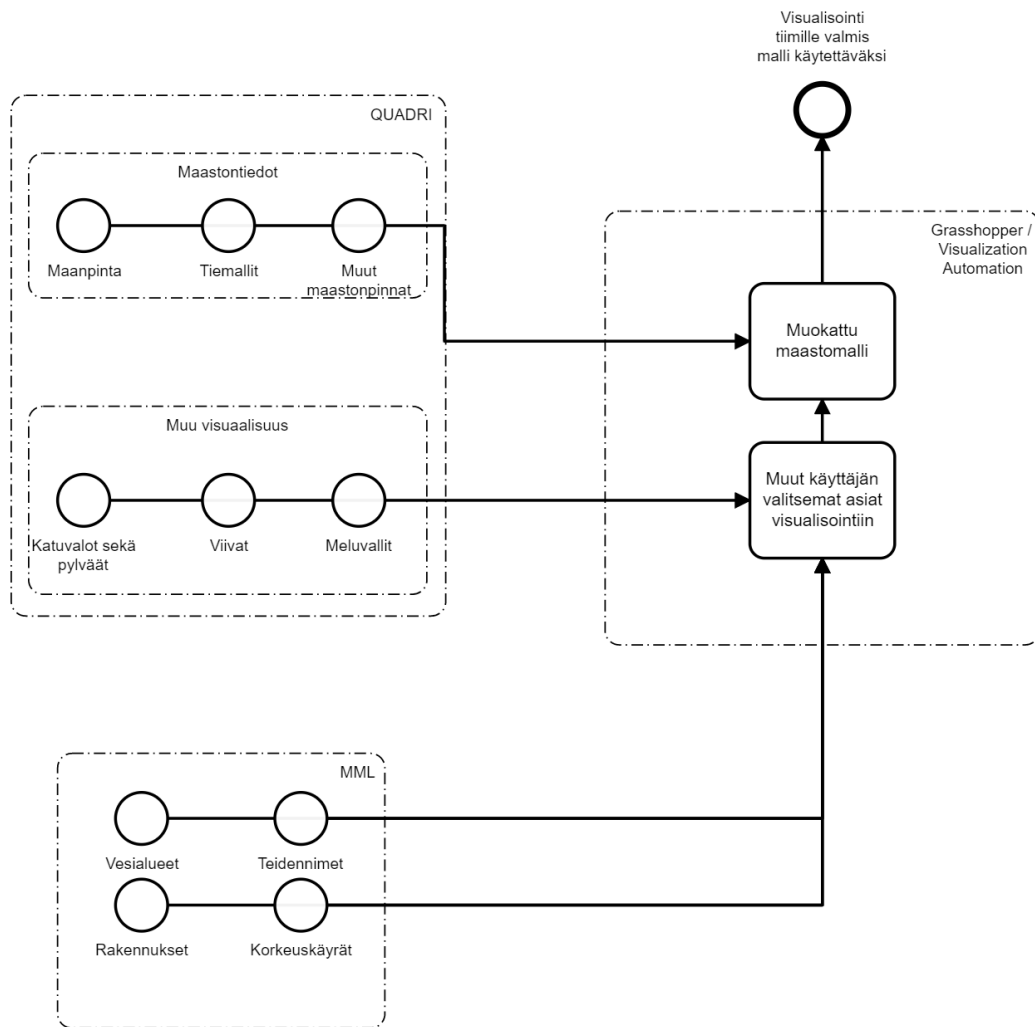
Kolmantena hyötynä Kilkelly (2016) ilmaisee automatisoinnin. Vaikka monilla laskennallisuus suunnittelun-sovelluksilla luodaan monimutkaista geometriaa, voidaan näillä sovelluksilla tehdä paljon muutakin. Sovellukset käyttävät eri järjestelmien rajapintoja ja monet näistä sovelluksista voivat suorittaa pitkäväteisiä työtehtäviä automaattisesti. Tällaisia tehtäviä ovat muun muassa erilaiset uudelleennimeämiset ja kopioinnit.

Neljäntenä asiana Kilkelly mainitsee erilaiset testausmahdollisuudet. Vaikka simulaatiosta saatava tieto ei täysin vastaa oikean elämän tapahtumia, on se parempaa kuin ei mitään. Laskennallisilla suunnittelu-sovelluksilla voit suorittaa laskennan siitä, paljonko aurinkoa suunnitelmasi rakennus saa puolipilvisenä maaliskuun päivänä.

Viimeisenä asiana artikkelissa ilmenee algoritminen ajattelu. Laskennallinen suunnittelu vaatii loogista askel askeleelta ajattelua. Kilkellyn (2016) mukaan useat arkkitehdit luottavat omaan intuitioon ja luovuuteen erilaisten ongelmien ratkonnassa, mutta tällainen ajattelu ei välttämättä toimi loogisen prosessin kanssa. Laskennallisella suunnittelulla voit koodata tämän intuition ja tarkastella ongelmaasi, sekä sen vaiheita, kunnes ymmärrät, mikä on ratkaisu ongelmaan. Lopuksi pystyt vielä parannella ja jatkokehittää näitä vaiheita sekä ratkaisua.

5 Rhinoceros, Grasshopper ja Quadri

Suunniteltavana oleva järjestelmä tulisi erittäin todennäköisesti rakentamaan käyttäen Rhinocerosin Grasshopper-sovellusta. Kuten luvussa kaksi todettiin, Grasshopperin tarjoamat mahdollisuudet erilaisten tiedostomuotojen tuomiseen ja viemiseen, mallien muokkaamiseen, sekä järjestelmän osien helppoon muokattavuuteen tekevät Grasshopperista järkevän valinnan järjestelmän päätyökaluksi. Myös Grasshopperin lisääntyvä käyttö Rambollin sisällä suosii Grasshopperin valintaa päätyökaluksi järjestelmälle.



Kuva 7. Järjestelmän tarvittavat tietovirrat

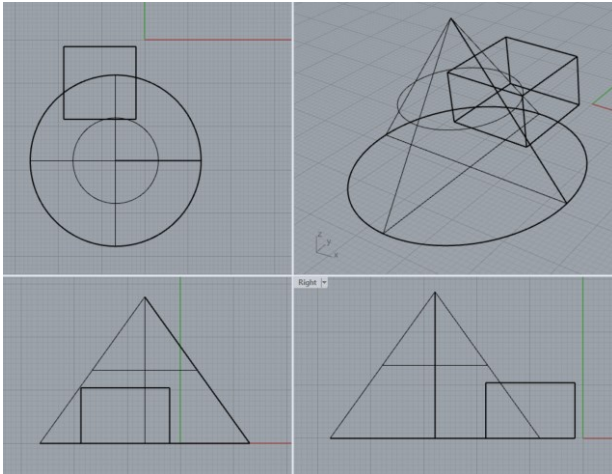
Vasemmalla kuvassa (kuva 7.) ovat eri tietolähteet, tässä tapauksessa Maanmittauslaitos sekä Quadri-sovellus. Näistä tietolähteistä otetaan tietoa Grasshopperiin, jossa tieto muokataan ja tehdään siitä malli.

Tällä hetkellä Quadri on se alusta, mihin kerätään erilaisten projektien kaikki suunniteltu tieto. Tästä syystä Quadri on valittu mahdollisen järjestelmän toiseksi pääsovellukseksi, vaikka tässä järjestelmässä Quadri toimisi ainoastaan tietolähteenä. Quadrin ongelmana on kuitenkin sama ongelma kuin maanmittauslaitoksen rajapinnassa: tietomäärä on erittäin suuri, eli jotenkin täytyisi valita mitkä tiedot valittaisiin 3D-malliin.

Tässä kappaleessa käymme läpi näitä kahta myös teoriaosuudessa mainittua sovellusta tarkemmin, tutkimme näiden sovellusten ominaisuuksia, mahdollisuuksia sekä erilaisia toimintoja, mitä sovelluksilla pystyttäisiin tekemään. Tässä kappaleessa tutkimaan myös erilaisia visuaalisia ongelmia, mitä Grasshopperissa voidaan korjata.

5.1 Rhinoceros näkymä ja käyttöliittymä

Rhinoceros, eli Grasshopperin pääsovellus on ulkonäöltään hyvin samankaltainen kuin muutkin 3D-malinnusohjelmat kuten 3DS Max tai Blender. Käyttäjällä on neljä eri kamera perspektiiviä, edestäpäin kuvaava, oikealta päin kuvaava, ylhäältä sekä käänneltävissä oleva kamera (kuva8.).



Kuva 8. Rhinoceros-sovelluksen perusnäkö

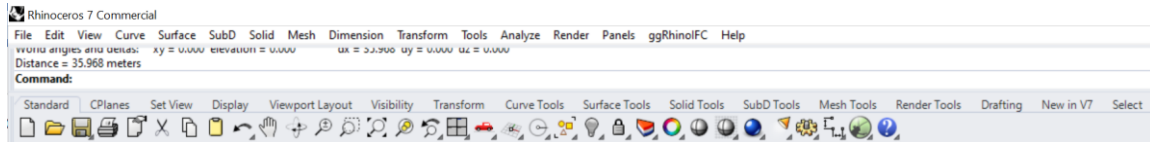
Kameraikkunoiden vasemmalla puolella on erilaisia toimintoja, kuten objektien muodostamista, alueen rajausta sekä erilaisia yhdistämis- ja erotustoimintoja (kuva 9.). Rhinocerosilla pystytään siis tekemään objektien muokkausta, luomista ja visuaalisia näkymiä täysin ilman Grasshopper-lisäosaa.



Kuva 9. Rhinoceros-sovelluksen vasemman puolen toiminnot

Näkymän yläreuna on myös vastaavanlainen kuin muissakin järjestelmissä (kuva 10.). Yläreunan painikkeista tapahtuvat tallennus, kopiointi ja leikkaus, sekä ehkä tärkeimpänä painikkeena vihreä Grasshopperin käynnistys painike. Yläpalkissa on myös tekstinsyöttö

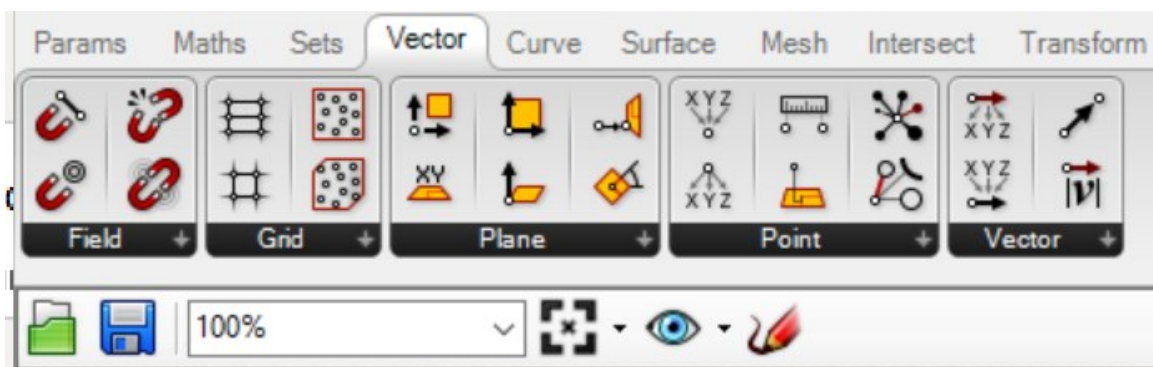
alue, johon voidaan kirjoittaa erilaisia komentoja esimerkiksi matkan mittaamiseen 3D maailmassa.



Kuva 10. Rhinoceros-sovelluksen yläpalkin toiminnot

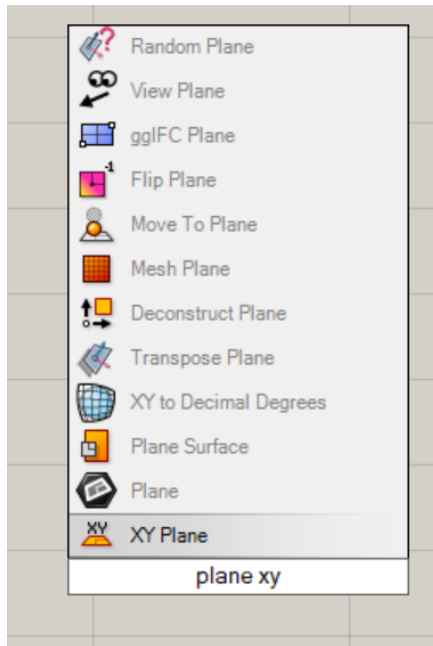
5.2 Grasshopper- näkymä ja käyttöliittymä

Grasshopperin näkymä on taas hyvinkin poikkeava perinteisistä suunnitteluohjelmista kuten CAD:ista. Näkymästä suurimman osan vie niin sanottu kanvaasi. Kanvaasi on se alue, mihin kaikki mahdolliset komponentit sijoitetaan. Komponentteja kanvaasille voidaan ottaa joko yläpalkin kautta valitsemalla ensiksi haluttu komponenttiryhmä, ja sitten itse komponentti (kuva 11.). Komponenttiryhmiä ovat esimerkiksi matemaattiset komponentit, viivaan liittyvät komponentit sekä erilaiset Grasshopperin lisäosat kuten Human UI tai Mosquito.



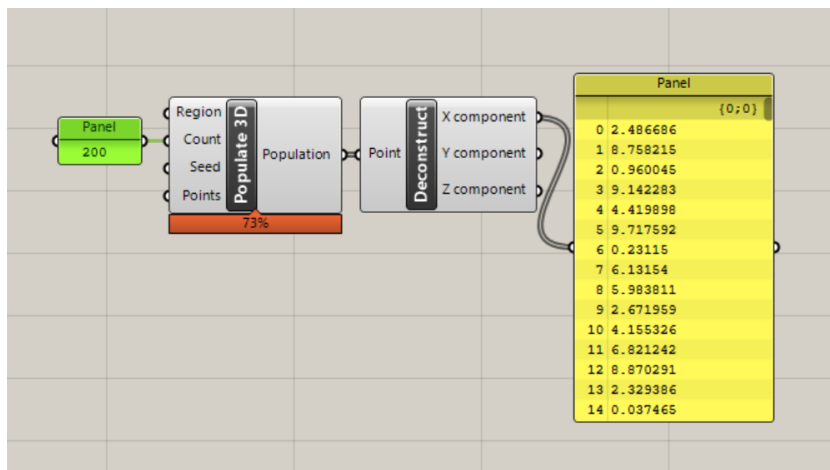
Kuva 11. Grasshopper Vector-ryhmän komponentteja

Toinen vaihtoehto komponenttien lisäämiseen on kaksoisnapsauttaa kanvaasin tyhjää aluetta ja aloittaa kirjoittamaan komponentin nimeä (kuva 12.).



Kuva 12. Kanvaasin kautta komponentin lisäys

Grasshopper pystyy myös näyttämään kanvaksella erilaisia tietoja esimerkiksi paneelien kautta. Paneeleita käytetään yleensä tiedon tarkastamiseen, mutta niillä voidaan myös viedä tietoa johonkin komponenttiin.



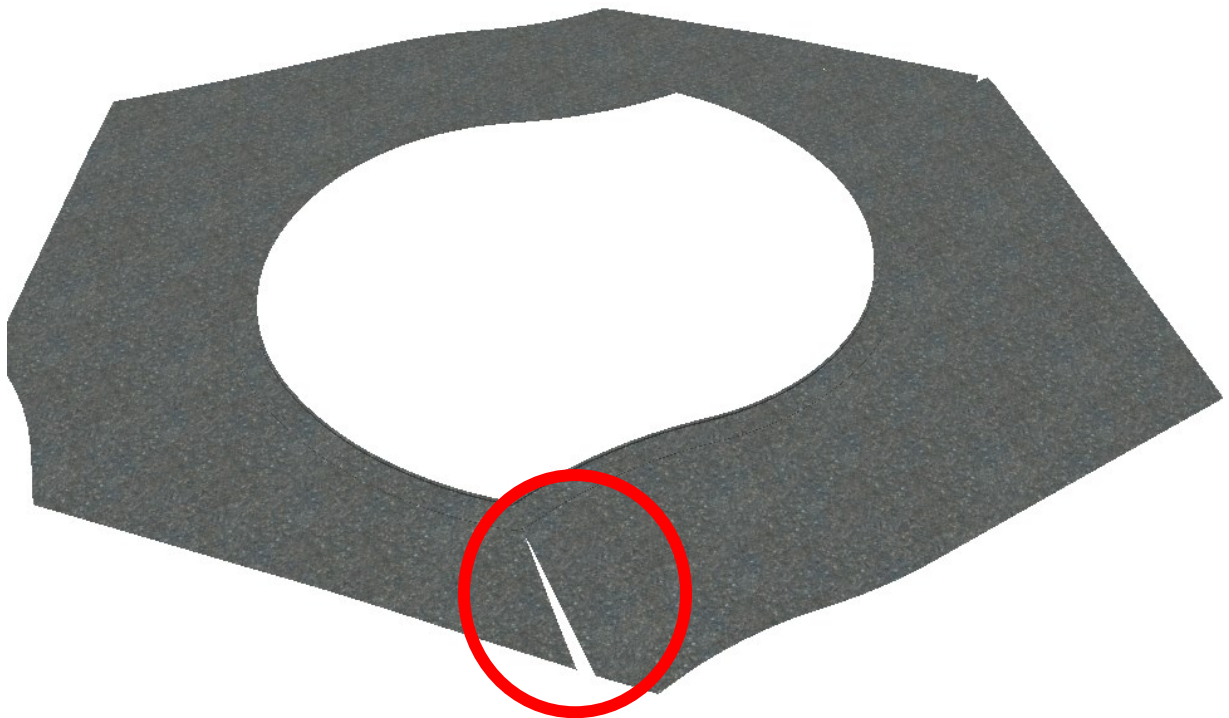
Kuva 13. Paneelien käyttö sisään- ja ulostulona

Kuvan 13. tilanteessa siis luodaan 200 kappaletta pisteitä, hajotetaan pisteet palasiin ja näytetään paneelissa jokaisen pisteen X koordinaatti.

5.2.1 Grasshopper esimerkki ongelmasta

Väyläsuunnittelijoiden mukaan suunnitteluohjelmissa on pieniä teknisiä ongelmia, jotka aiheuttavat Quadriin ongelmia. Yksi tällainen ongelma on alue, johon ei ole pystytty muodostamaan vaadittua pintaa teknisten syiden johdosta. Tämän tyyppisiä ongelmia voi siis olla useampi, joten olisi tärkeää, jos mahdollinen järjestelmä automaattisesti tarkastaisi ja korjaisi vastaavat ongelmat Grasshopperin puolella.

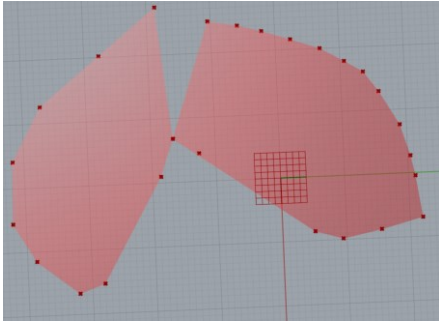
Suunnitteilla olevassa väylämallissa näyttää kaikki olevan kunnossa. Tarkemmalla tutkimalla löydetään, että liikenneympyrään kaistojen muodostuksessa on syntynyt reikä yhden liittymän kohdalle (kuva 14.).



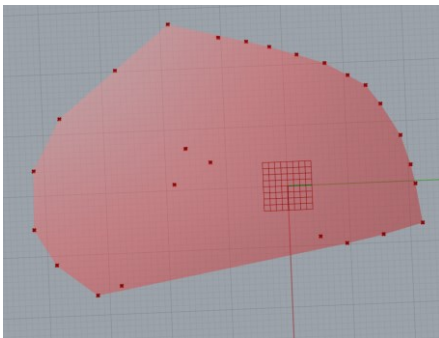
Kuva 14. Quadriissa näkymä toiminnallinen ongelma

Järjestelmän täytyisi siis pystyä tunnistamaan vastaavanlaiset virheet ja muodostaa siihen automaattisesti puuttuva pala. Grasshopperin monilla työkaluilla kyseinen ongelma

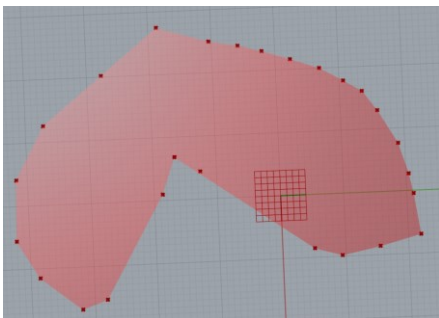
voitaisiin ratkaista, mutta tämä taas tuo uuden ongelman järjestelmään: miten voidaan tietää mitkä aukot todella kuuluvat suunnitelmaan ja mitkä eivät. Mikäli kaikki aukot paikataan, yläpuolella olevan kuvan liikenneympyrä umpeutuu kokonaan.



Kuva 15. Ongelma



Kuva 16. Virhe automaatio



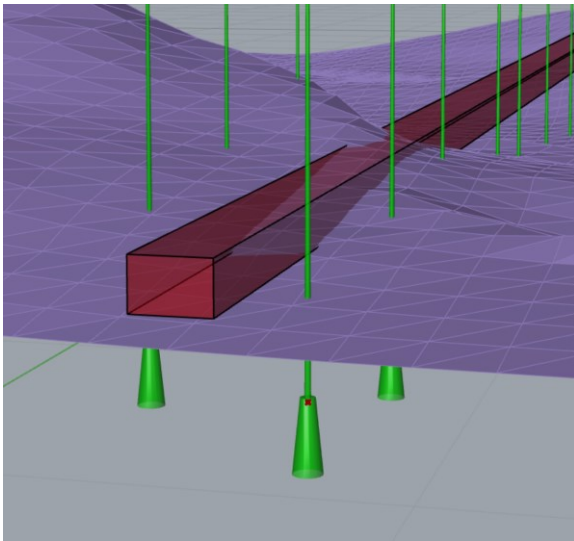
Kuva 17. Onnistunut versio

Ensimmäisessä kuvassa on ongelmatilanne, joka täytyisi korjata (kuva 15.). Toisessa kuvassa on esimerkki mitä voi tapahtua, jos automatisointi menee pieleen (kuva 16.).

Viimeisessä kuvassa on onnistunut versio tilanteesta (kuva 17.). Kuvan esimerkit eivät ole automatisoitu, vaan käsin tehtyjä esimerkkejä mahdollisesta ongelmasta.

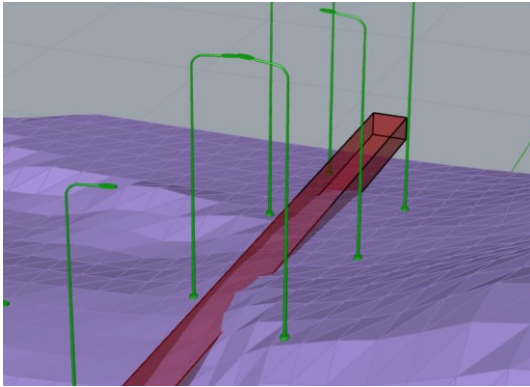
5.2.2 Grasshopper-objektit maanpintaan ja oikeaan suuntaan

Visualisointiryhmällä on ollut ongelmia lisätä katuvalaisimia visuaaliseen malliin. Tämä ongelma pystytään Grasshopperin automatisoinnilla korjaamaan muutamalla komponentilla. Ongelmana on ollut, että katuvalaisimet saattavat olla metrin maanpinnan päällä tai alla, ja osoittavat väärään suuntaan (kuva 18.). Grasshopperin täytyy siis muokata katuvalaisimia niin, että ne ovat visuaalisesti oikeassa sijainnissa ja kääntyneenä oikeaan suuntaan.



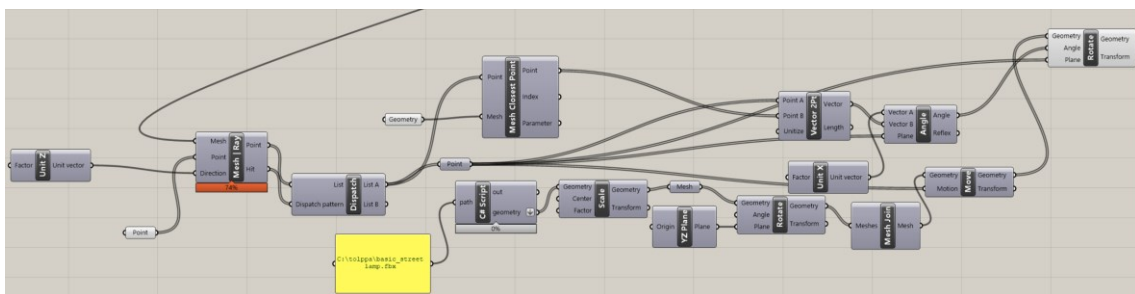
Kuva 18. Lamppujen epäonnistunut sijoittaminen

Nykytilanteessa visualisointiryhmä on käsin siirtänyt kuvassa olevat katuvalaisimet kohdilleen. Automatisoinnilla tämä vaihe saataisiin korjattua hyvin helposti. Kuvassa 19. valaisimet ovat nyt kuin ne olisivat oikeasti asennettuna maastoon. Maan alla on sinne suurimmaksi osaksi kuuluva betoniperusta ja valaisin osa on tietä edustavan laatikon puolella.



Kuva 19. Lamppujen sijaintien onnistunut automatisointi

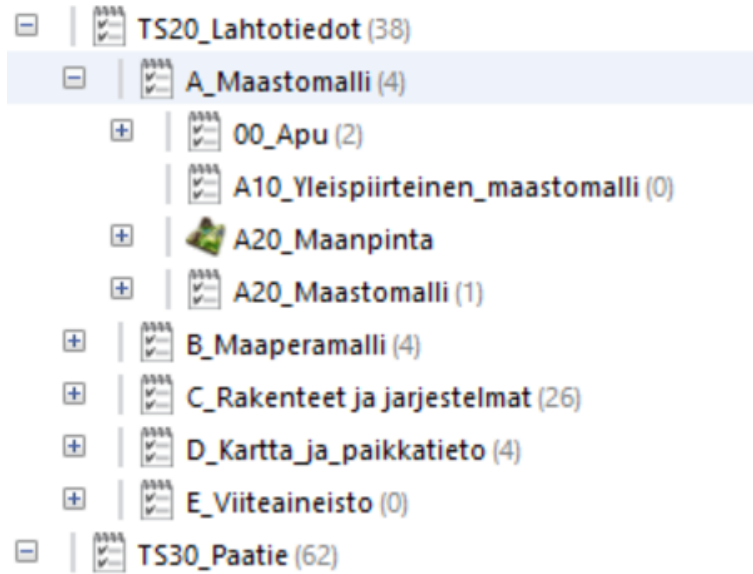
Tämä onnistunut automatisointi on tehty muutamilla yksinkertaisilla Grasshopperin komponenteilla ja voisi näin olla osa visualisointijärjestelmää (kuva 20.). Automatisointi tarkastelee lamppujen toivottuja sijoituspisteitä, ja näiden pisteiden kohdalla laukaisee säteen kohtisuoraan maastomallia kohti. Sijainti, jossa säde osuu maastomalliin, otetaan talteen. Tallennettuihin osumakohtiin sijoitetaan katuvalaisimet, jonka jälkeen ne käännetään osoittamaan kuvitteellista tietä kohti.



Kuva 20. Onnistunut automatisointi

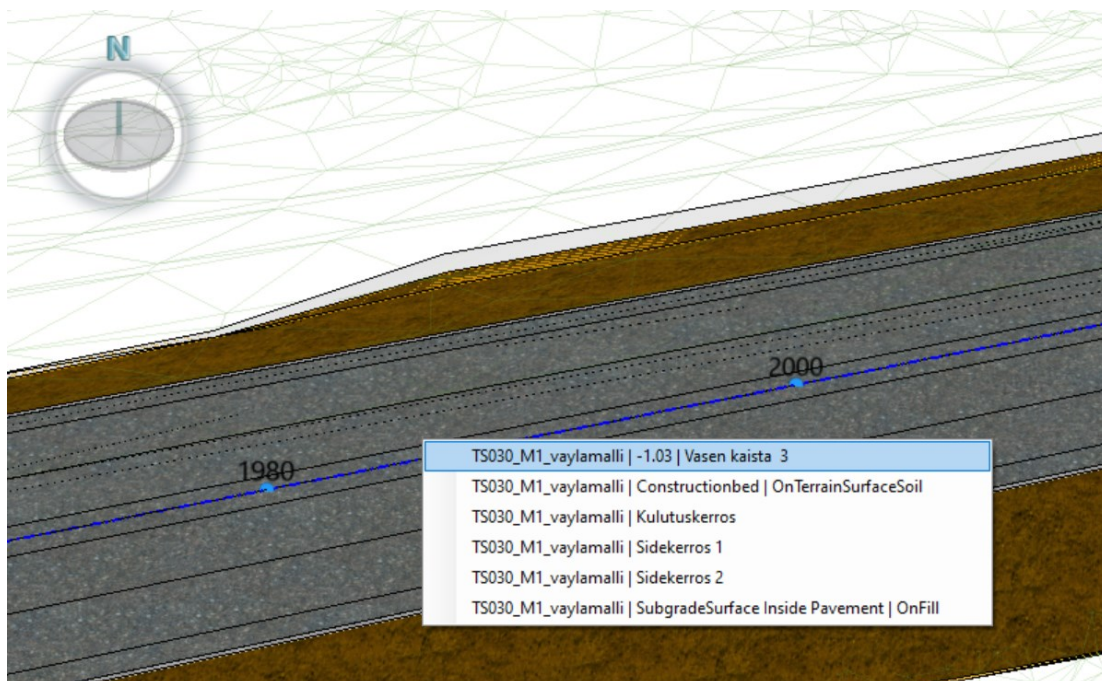
5.3 Quadri-näkymä

Quadrin näkymä on myös hyvin perinteinen. Vasemmalla on niin sanottu selainikkuna (kuva 21.). Selainikkuna sisältää kyseisen projektin kaikki tiedot, niin alkutiedoista suunnittelutietoihin, sekä pienempiin yksityiskohtiin kuten valaisimien sijainteihin.




Kuva 21. Quadri selain ikkunan näkymä

Keskellä on 3D-malli selaimessa valituista tiedoista ja tätä mallia pystytään katselemaan eri kulmista (kuva 22.). Quadri käyttäjä voi valita yhden objektin ja tarkastella sen ominaisuuksia oikealla sijaitsevasta ominaisuudet ikkunasta.



Kuva 22. Quadrissa oleva tie

Ominaisuudet ikkunasta käyttäjä näkee objektin ID:n, nimen ja mihin isompaan malliin se kuuluu (kuva 23.).

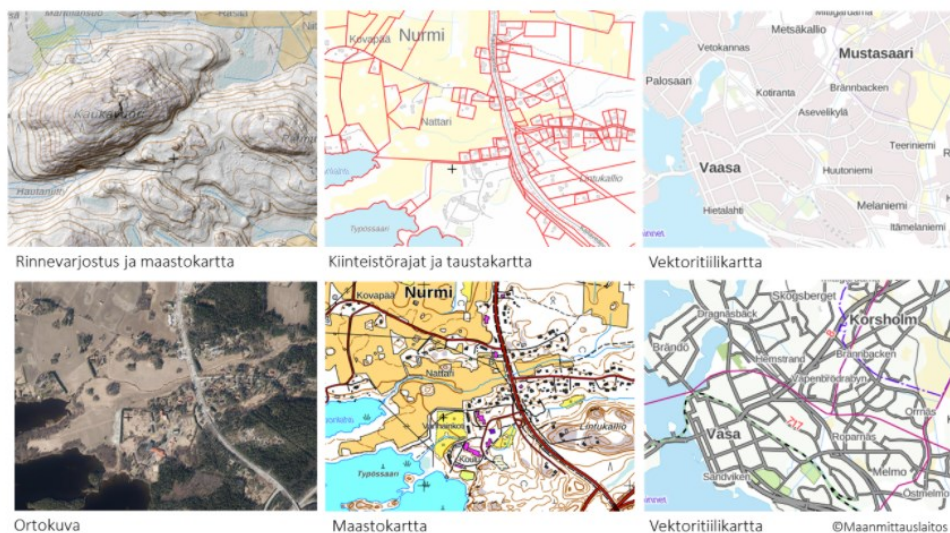
Ominaisuudet	
 Kaista	
<input checked="" type="checkbox"/> Sisäiset attribuutit	
Ominaisuus ID	557710
Ominaisuustyyppi ID	1310033
Ominaisuuden tila	ReadOnly
Versio ID	1
GUID	8c9d8f76-1477-4e2b-9897-2bdaf2aac92e
Feature Size (bytes)	154944
<input checked="" type="checkbox"/> Tunniste	
Nimi	TS030_M1_vaylamalli -1.03 Vasen kaista 3
Suunniteltu/Nykyinen	Suunniteltu [1]
<input checked="" type="checkbox"/> Classification	
<input checked="" type="checkbox"/> Tehtävät	
Tehtävä	TS030_M1_vaylamalli

Kuva 23. Quadrin valitun objektin ominaisuudet ikkuna

6 Järjestelmän mahdolliset rajapinnat ja maanmittauslaitoksen rajapinta

Hankealueelle ominaisia tietoja voidaan kuvata rajapinnasta saatavien tietojen avulla. Tiedot kuvaavat olemassa olevaa nykytilannetta ja näin ollen auttavat hahmottamaan suunniteltuja malleja nykyisen ympäristön avulla. Avoimia rajapintoja ovat esimerkiksi Maanmittauslaitoksen rajapinta, SYKE, avoindata ja Väyläviraston avoin rajapinta. Nämä kaikki rajapinnat voivat sisältää visualisoinnin kannalta arvokasta tietoa, mutta tässä tutkimuksessa tarkastellaan tarkemmin Maanmittauslaitoksen rajapintaa.

Maanmittauslaitoksen (2021) kartat sisältävät paikannimiä, teitä, osoitteita, rakennuksia, hallintorajoja sekä maastonkuvioita ja korkeuksia. MML:n rajapintapalveluista saatavat aineistot soveltuvat heidän mukaansa yhdistettäväksi muihin aineistoihin, eli esimerkiksi suunnitteilla olevaan järjestelmään. Maanmittauslaitoksen (2021) tuotteet perustuvat maastotietokantaan, ortoilmakuviin, nimistörekisteriin ja näistä johdettuihin pienempiin aineistoihin. Näitä tuotteita on saatavilla erilaisilla rajapintatekniikoilla, joita ovat WMS, WMTS sekä vektoritiilet (kuva 24.).



Kuva 24. Maanmittauslaitoksen tarjoilu

Maanmittauslaitokselta saatavassa tiedossa on paljon hyvää liitännäistietoa, eli metatietoa kuten kuvassa 25. ilmenee esimerkiksi rakennuksista ja teistä kuvassa 26. Jos tätä metatietoa voitaisiin käyttää hyväksi tiedon automatisoinnissa, mallin uskottavuus kasvaisi. Maanmittauslaitoksen maastotietokohteet (2018) tiedostossa on paljon tietoa esimerkiksi rakennuksista (kuva 25.). Jos rakennusten korkeustietoja pystyttäisiin käyttämään hyväksi visualisoinnissa, tämä lisäisi todenmukaisuuden tunnetta.

Liike- tai julkinen rakennus, ? krs	Rakennukset	Viiva Alue	27 75	42120 42220
Liike- tai julkinen rakennus, 1-2 krs	Rakennukset	Viiva Alue	27 75	42121 42221
Liike- tai julkinen rakennus, 3-n krs	Rakennukset	Viiva Alue	27 75	42122 42222
Lomarakennus, ? krs	Rakennukset	Viiva Alue	27 75	42130 42230
Lomarakennus, 1-2 krs	Rakennukset	Viiva Alue	27 75	42131 42231
Lomarakennus, 3-n krs	Rakennukset	Viiva Alue	27 75	42132 42232
Teollinen rakennus, ? krs	Rakennukset	Viiva Alue	27 75	42140 42240
Teollinen rakennus, 1-2 krs	Rakennukset	Viiva Alue	27 75	42141 42241
Teollinen rakennus, 3-n krs	Rakennukset	Viiva Alue	27 75	42142 42242
Kirkko	Rakennukset	Viiva Alue	27 75	42170 42270
Kirkollinen rakennus, ? krs	Rakennukset	Viiva Alue	27 75	42150 42250
Kirkollinen rakennus, 1-2 krs	Rakennukset	Viiva Alue	27 75	42151 42251
Kirkollinen rakennus, 3-n krs	Rakennukset	Viiva Alue	27 75	42152 42252
Muu rakennus, ? krs	Rakennukset	Viiva Alue	27 75	42160 42260
Muu rakennus, 1-2 krs	Rakennukset	Viiva Alue	27 75	42161 42261
Muu rakennus, 3-n krs	Rakennukset	Viiva Alue	27 75	42162 42262

Kuva 25. Maanmittauslaitoksen tietojen metatietoja

Taulukon tiedoissa visualisoinnin kannalta hyödyllisiä ovat kohdeluokan-nimi ja kohde-luokkanumero eli viimeinen sarake. Näiden tietojen perusteella pystyttäisiin

automatisoimaan esimerkiksi teolliset rakennukset betonimateriaalilla, ja niille määritettyjen kerroslukujen mukaan.

Tieviiva	Tiestö	Viiva	25	12100	Tiesymboli, Tieteksti
Autotie Ia	Tiestö	Viiva	25	12111	
Autotie Ib	Tiestö	Viiva	25	12112	
Autotie IIa	Tiestö	Viiva	25	12121	
Autotie IIb	Tiestö	Viiva	25	12122	
Autotie IIIa	Tiestö	Viiva	25	12131	
Autotie IIIb	Tiestö	Viiva	25	12132	

Kuva 26. Maanmittauslaitoksen tietojen metatietoja

Myös autoteistä on samanlainen jaottelu tarjolla, ja samasta maastotietokanta tiedostosta löytyy selitys näille kohdeluokille (kuva 27.). Maanmittauslaitoksen maastotietokanta tiedoston mukaan, tien kohdeluokka määräytyy sen keskikunnon perusteella, joten lyhyitä keskimääräisesti parempia tai huonompia kohteita ei erotella.

Luokkien määrittely:

Haja-asutusalueen tiet ja kadut:

Autotie Ia: moottoritien kaksi- tai useampikaistainen ajorata.

Autotie Ib: muun kaksiajorataisen kuin moottoritien kaksi- tai useampikaistainen ajorata tai yksiajoratainen, kaksi- tai useampikaistainen autotie, ajoradan leveys on yli 8 m.

Autotie IIa: yksiajoratainen, kaksikaistainen, ajoradan leveys on 6,5 - 8 m.

Autotie IIb: yksiajoratainen, kaksikaistainen, ajoradan leveys on 5 - 6,5 m.

Autotie IIIa: yksiajoratainen, yksikaistainen, ajoradan leveys on 4 - 5 m.

Autotie IIIb: yksiajoratainen, yksikaistainen, ajoradan leveys on 3 - 4 m.

Ajotie: yksiajoratainen, yksikaistainen, ajoradan leveys on alle 3 m.

Kuva 27. Maanmittauslaitoksen tietojen selitys

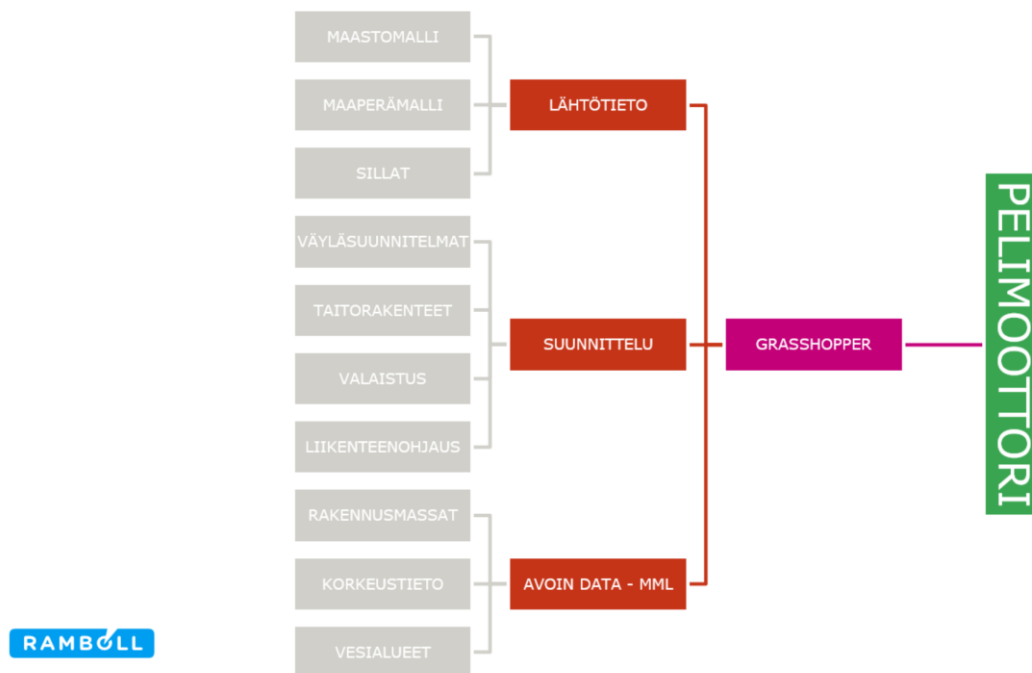
Paikkatietojen rajapintapalveluiden (2021) kautta käyttäjä saa käyttöönsä Maanmittauslaitoksen ylläpitämiä paikkatietoja. Näitä paikkatietoja ovat muun muassa tiet, rakennukset, vesistöt ja paikannimet. Paikkatietojen rajapintapalvelut sopivat käyttäjän käyttöön, kun käyttäjä haluaa hakea sovellukseen tai järjestelmäänsä yksittäisten kohteiden tietoja kuten osoitteita tai paikannimiä. Jotta näitä tietoja voitaisiin hakea, käyttäjällä täytyy olla

ohjelmisto, joka tekee hakupyynnöt Maanmittauslaitoksen palvelimelle. Tällaisia ohjelmistoja voivat olla esimerkiksi paikkatieto-ohjelmistot sekä itse ohjelmoidut sovellukset.

7 Suunnitelma järjestelmälle

Järjestelmän tärkeimpiä ominaisuuksia olisi sen monikäyttöisyys erilaisissa tilanteissa, mahdollisimman pitkälle viety automatisointi, joka kuitenkin tarjoaisi käyttäjälle valintoja, sekä yksinkertainen, selkeä ja helppokäyttöinen käyttöliittymä. Järjestelmällä täytyisi pystyä valitsemaan erilaisia tietolähteitä niin maanmittauslaitoksen rajapinnasta, käytössä oleviin tiedostomuotoihin ja tärkeimpänä, sen täytyisi pystyä jotenkin käyttämään suurta tietoaimeistoa, kuten Quadria. Erilaisten tietojen esittäminen käyttäjälle esimerkiksi käyttöliittymän välityksellä, lisäisi järjestelmän käytettävyyttä. Kuvassa 28. on yksinkertaistettu kuva järjestelmästä ajatustasolla.

VISUALISOINNIN AUTOMAATIO



Kuva 28. Yksinkertainen kuva järjestelmästä

7.1 Haastattelut

Tutkimuksen alussa suoritettiin haastatteluja eri tekniikka-alojen ammattilaisille. Haastatteluiden perusteella saatiin selville tekniikka-alojen käyttämiä sovelluksia, sekä näiden sovellusten rajapintoja. Haastatteluissa haluttiin selvittää tekniikka-alojen pääsovellukset, sekä näiden sovellusten mahdolliset rajapinnat Grasshopperiin. Viimeseinä kysymyksenä haastattelussa esitettiin vielä yleisimpiä haasteita tekniikka-alan sovellusten ja Grasshopperin rajapintojen käytössä.

Haastatteluiden perusteella ilmeni, että haasteita on muun muassa koordinaatiston kanssa. Tämä ongelma ilmenee mallissa siten, että objektit muodostuvat väärin sijainteihin. Toisena ongelmana maisema suunnittelija Kemppainen (Teams palaveri 04.01.2021) mainitsi erilaiset mallien materiaaliongelmat, esimerkiksi puuttuvat materiaalit joistain objekteista.

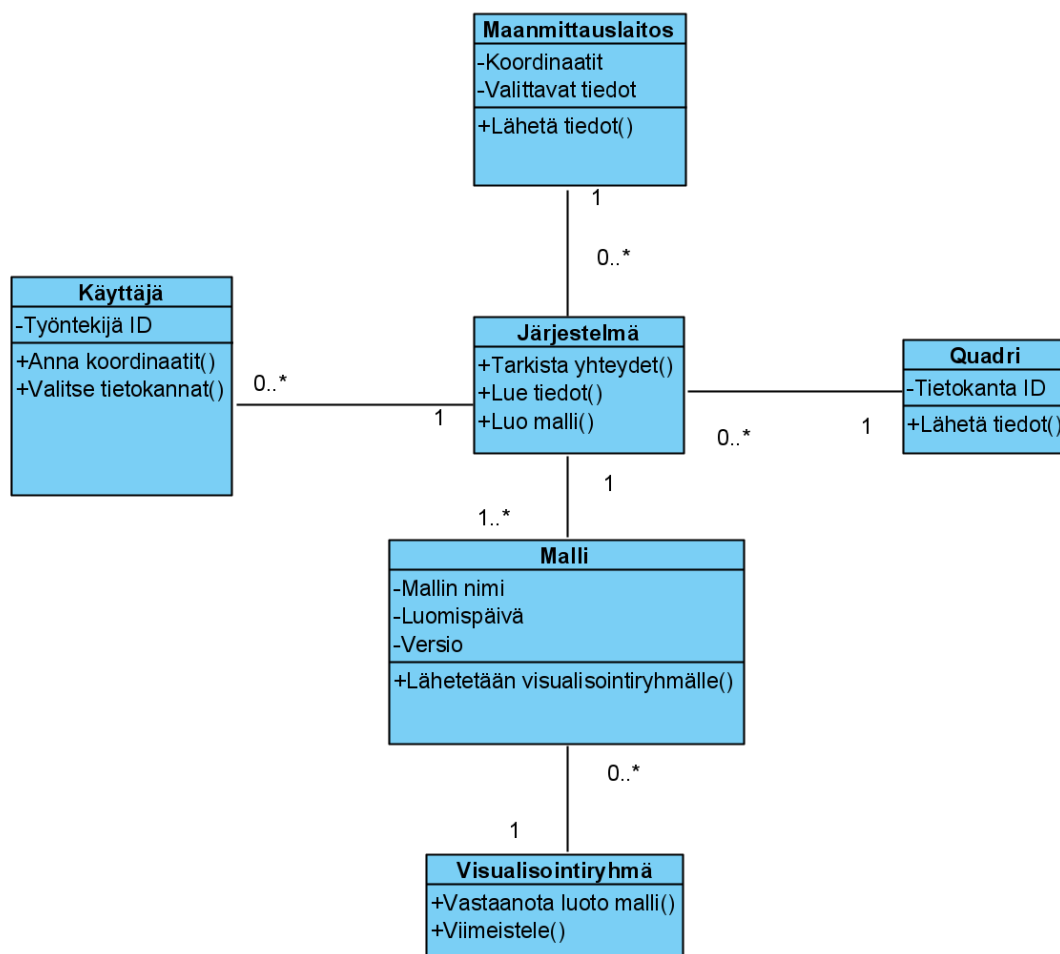
Haastatteluja suoritettiin myös pelimoottorin kanssa työskentelevälle visualisointiryhmälle. Visualisointiryhmän kertomia ongelmia on tutkittu kappaleessa ”Grasshopper-näkymä ja käyttöliittymä” ja tässä kappaleessa on myös osaksi ratkottu heidän ongelmiansa.

7.2 UML kuvaukset

Tehdyt UML kaaviot auttavat järjestelmän suunnittelussa, sillä niistä saa hyvin selkeän kuvan, kuinka järjestelmä toimisi ja mitä riippuvaisuuksia se sisältää. Luokkakaavion avulla voidaan taas huomata erilaisia riippuvuuksia eri luokkien välillä, sekä havaita mahdollisia ongelmakohtia. Tehdyt kaaviot ovat kuitenkin hyvin suppeita ja tarkemmat kaaviot on syytä tehdä, jos oikea järjestelmä rakennetaan.

7.2.1 Luokkakaavio kuvaus

Luokkakaaviosta voimme tutkia suunniteltua järjestelmää ja eritoten sen riippuvaisuuksia (kuva 29.). Keskeltä aloittaessa voimme huomata, että järjestelmä on yhteydessä yhteen maanmittauslaitokseen, yhteen Quadrin tietokantaan, nollasta yhteen käyttäjään eli järjestelmä voi olla käynnissä, vaikka työntekijä ei olisikaan paikalla ja yhdestä n:ään malliin, eli samasta lähtöaineistosta voidaan tuottaa useampi malli erilaisilla käyttäjän valinnoilla.



Kuva 29. Järjestelmän yksinkertainen esimerkki luokkakaavio

Luokkien sisältämät attribuutit ovat vain esimerkkejä tiedoista mitä kyseisissä luokissa voisi olla. Erilaiset ID numerot auttavat mahdollisissa ongelman selvitystilanteissa, sekä

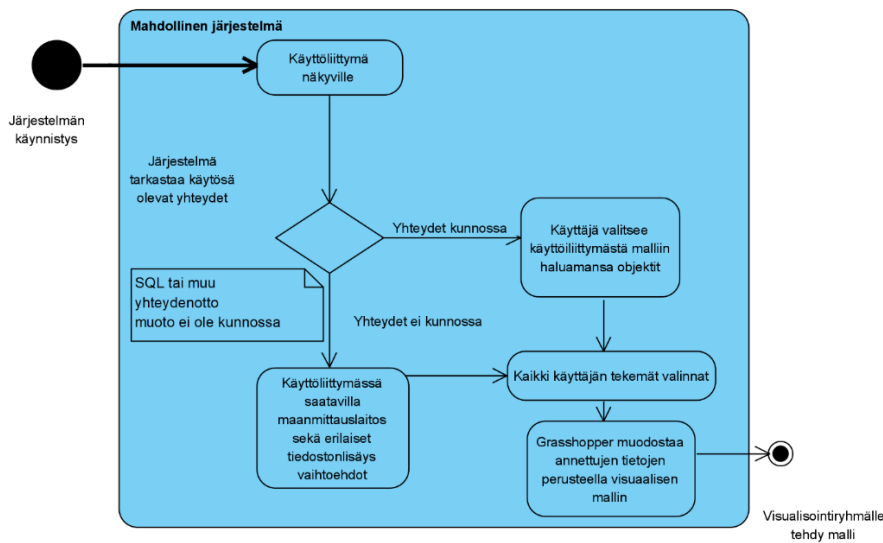
lisäävät selkeyttä tilanteissa, jossa esimerkiksi monta työntekijää on työstänyt samaa mallia järjestelmällä (kuva 30.). Muuten hyödyllisiä tietoja voivat olla esimerkiksi mallin luomispäivä sekä mikä mallin versio on kyseessä.

Käyttäjä
-Työntekijä ID
+Anna koordinaatit()
+Valitse tietokannat()

Kuva 30. Luokkakaavion "käyttäjä" luokan esittely

7.2.2 Aktiviteettikaavio kuvaus

Kuvan 31. aktiviteettikaaviosta näemme kuinka järjestelmä voisi toimia teoriassa. Käyttöliittymä näkyisi käyttäjälle ja samalla järjestelmä tarkistaisi onko erilaiset yhteydet, esimerkiksi SQL yhteys Quadrin tietokantaan kunnossa. Jos yhteys olisi kunnossa, käyttäjä saisi valita haluamansa tiedot tietokannasta sekä maanmittauslaitoksen rajapinnasta. Lopuksi kaikkien käyttäjän valitsemien vaihtoehtojen perusteella muodostettaisiin 3D-malli, joka voitaisiin antaa visualisointiryhmälle työstettäväksi.



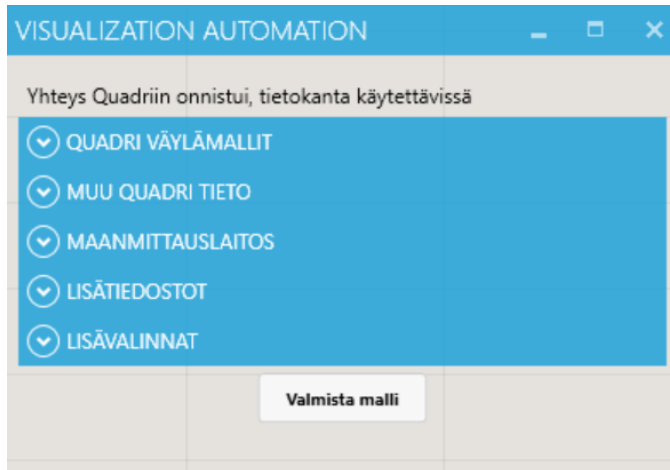
Kuva 31. UML-aktiviteettikaavio järjestelmästä

7.3 Käyttöliittymä esimerkki

Järjestelmän käyttöliittymä on yksi koko järjestelmän tärkeimmistä osista, sillä se on lopputuloksen kanssa ainoa asia mikä näkyy käyttäjälle. Siksi erityisesti käyttöliittymää on hyvä suunnitella ja kerätä mielipiteitä sen käyttäjiltä, että muutoksia pystyttäisiin jälkeenpäin tekemään. Käyttöliittymän täytyisi olla mahdollisimman selkeä ja yksinkertainen, sillä kaikki mahdollinen laskentateho tarvitaan mallin luomiseen.

Eri tietolähteet voitaisiin erotella omilla alasvetovalikoilla tai välilehdillä käyttöliittymässä kuten kuvassa 32. Loogista olisi edetä joko ylhäältä alaspäin tai vasemmalta oikealle, sillä tällainen eteneminen lisää myös käytettävyyttä kuten kappaleessa kaksi todettiin. Järjestyksenä voisi olla väylämallit, muu Quadrista tuleva materiaali, Maanmittauslaitoksen tiedot, muut tiedostomuodot ja viimeisenä erilaiset lisävalinnat Quadrin tietoihin, kuten minkälaisia valaisimia tiettyihin pisteisiin asetettaisiin. Näiden jälkeen voisi mahdollisesti tulla vielä jonkinlainen koonti kaikista valinnoista mitä käyttäjä on tehnyt, sekä aika-arvio kauanko mallin tekemisessä kuluu. Käyttöliittymän alareunassa olisi vielä

syytä olla painike käyttäjälle, mistä sitten mallin voi luoda sekä seuraavana aukeava ikkuna mistä valita tiedostomuoto sekä kansiosijainti.



Kuva 32. Laadittu UI-käyttöliittymä pienempänä

Käyttöliittymän ominaisuuksina olisi hyvä olla valinta mitkä väylämallit tulevat mukaan visualisointiin (kuva 33.). Väylämalli sisältää paljon erilaisia objekteja kuten teitä, viivoja ja maastonmuokkauksia, joten käyttäjällä voisi myös olla mahdollisuus valita näistä vain tietty osa tai kaikki yhdellä napin painalluksella.

Yhteys Quadriin onnistui, tietokanta käytettävissä

QUADRI VÄYLÄMALLIT

Valitse haluamasi väylämallit

- ☒ Valitse kaikki
- ☐ Väylämalli 1
- ☐ Väylämalli 2
- ☐ Väylämalli 3
- ☒ Väylämalli 4
- ☒ Väylämalli 5
- ☐ Väylämalli 6
- ☐ Väylämalli 7

MUU QUADRI TIETO

MAANMITTAUSLAITOS

LISÄTIEDOSTOT

LISÄVALINNAT

Quadri

Olet tehnyt seuraavat valinnat:

Väylämallit 1, 4 ja 5

Valaisinsijainnit 1,2,3,4 ja 5

Maanmittauslaitos

Olet tehnyt seuraavat valinnat:

Rakennukset Tiet Vesistöt

Lisätiedostot

Olet tehnyt seuraavat valinnat:

IFC tiedostot 2 kpl

LandXML tiedostot 1 kpl

Valaisinpisteet 1,3,4,5 mallilla lamppu1

valaisinpisteet 2 mallilla lamppu2puolinen

Valmistamalli

Arvioitu aika 8 minuuttia

Kuva 33. Laadittu UI-käyttöliittymä kokonaan

7.4 Hyödyllisiä ominaisuuksia

Järjestelmällä olisi tärkeää olla erilaisia tarkastusmetodeja. Tietolähteiden väliset koordinaatit voivat olla eri muodossa, joka aiheuttaisi objektien ilmestymisen väärään sijaintiin. Lisäksi järjestelmä voisi tarkistaa mitä yhteyksiä sillä on saatavilla, onko sillä mahdollisesti toimiva yhteys Quadrin tietokantaan, tai mitä tietoja on saatavilla Maanmittauslaitokselta tiettyjen koordinaattien sisällä. Nämä erilaiset tarkastusmenetelmät vähentäisivät virheiden mahdollisuuksia.

Olisi tärkeää rakentaa järjestelmä niin, että se antaa käyttäjälle palautetta, niin valituista objekteista, mahdollisista tietolähteistä kuin erilaisista aika-arvioista kuinka kauan mallin tai osamallin tekemiseen saattaisi kulu. Jos järjestelmä antaisi palautetta käyttäjälleen, järjestelmän käytettävyys lisääntyisi, mutta käyttäjä myös osaisi arvioida omaa työpanostustaan ja käyttää odotteluun kuluvan ajan järkevämmiin, kuin istuen koneen ääressä. Annettua palautetta voisi myös olla erilaiset tekstitiedostot, kuten objektien ns. keski-kohta pisteet. Tällaiset keskikohta pisteet voivat tulla tarpeen myöhemmin, jos muita objekteja täytyy lisätä malliin esimerkiksi Unreal Enginessä.

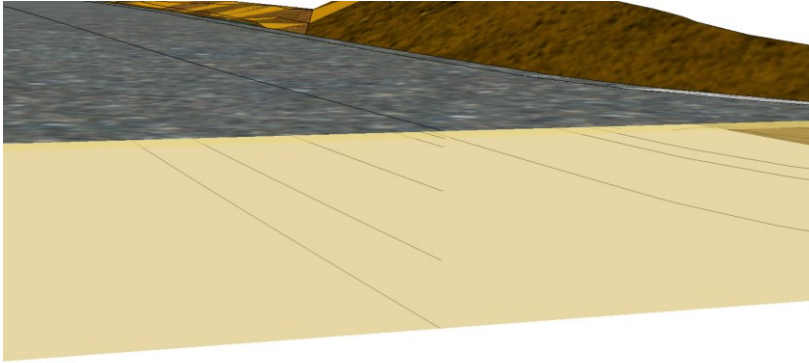
Tärkein ominaisuus järjestelmällä olisi kuitenkin luoda hyvälaatuinen malli, joka sisältää kaiken tarvittavan. Mallissa täytyisi olla suunnitellut väylämallit, eli tiet ja maanpinnan muutokset sekä sillat ja erilaiset muut visuaalisesti tärkeät ominaisuudet kuten meluvalit ja reunakivet. Visuaalisesti ei tärkeitä asioita olisi myös hyvä sisällyttää malliin mukaan. Tällaisia asioita olisi muun muassa mittalinjat, joiden mukaan tiet kulkisivat. Näitä mittalinjoja käyttäjä ei siis näe, mutta niitä voidaan tarvita vielä Unreal Enginen puolella.

7.5 Haasteet

Suurimmat haasteet järjestelmän rakentamisen kannalta ovat tietojen saaminen Quadrista sekä maanmittauslaitoksen rajapinnan kautta. Rambollin kokemuksen perusteella maanmittauslaitoksen rajapinta on erittäin hidas, eli suurien tietomäärien kuten rakennusten lataaminen sen kautta on aikaa vievää. Quadrista tietoa voidaan saada monella tapaa, rajapinnan kautta tämä yhteys olisi kenties viisainta tehdä, näin välttyttäisiin turhilta tiedostojen kopioimisilta ja siirtämiseltä.

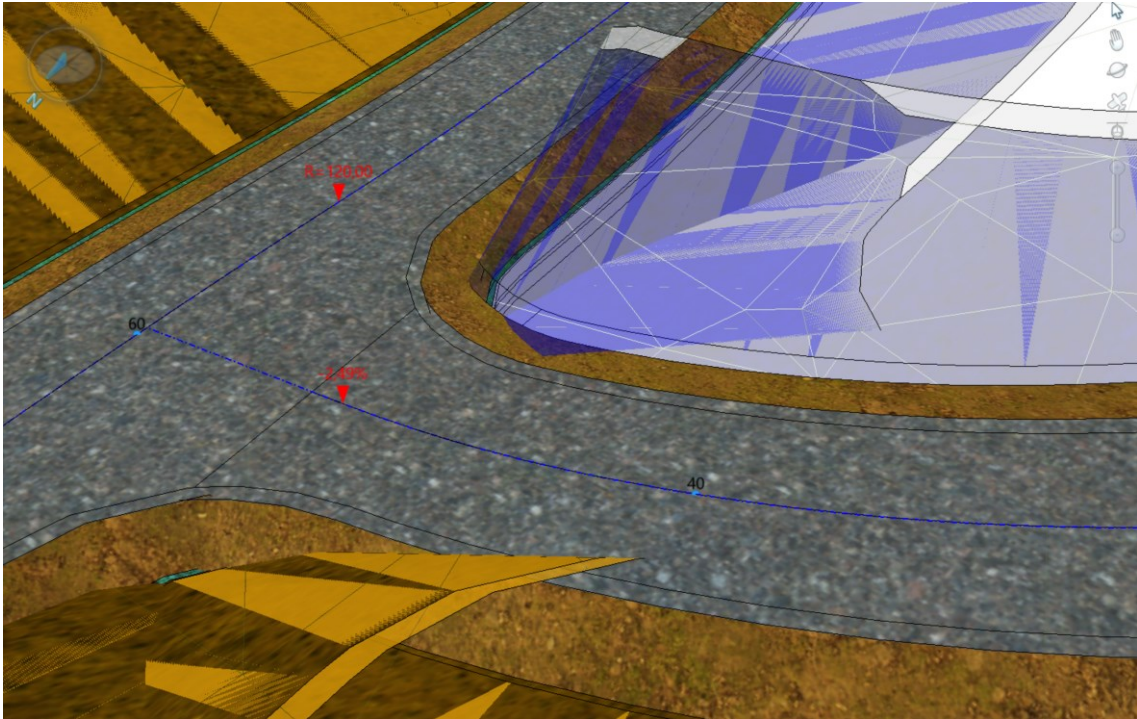
Haasteita syntyy myös maastonmuokkauksen yhteydessä. Quadrin sisältämässä tiedossa on erittäin paljon ylimääräistä materiaalia ja tätä materiaalia täytyisi pystyä automaattisesti muokkaamaan niin, että mallissa on vain visuaalisesti "välttämätön" tieto.

Quadrissa otetussa kuvassa 34. näemme poikkileikkauksen tiestä. Tästä tiestä täytyisi saada vain suoraan yläpuolelta näkyvät objektit, eli tien- ja maanpinta.



Kuva 34. Quadrin sisältämä tie poikkileikkaus

Toinen Quadrin ongelma, joka siirtyy samalla Grasshopperin puolelle on mallien päällekkäisyys. Jos tietoja otetaan Quadrasta Grasshopperiin, tulee tämä ongelma sinne myös mukana, ja tällaiset päällekkäisyydet täytyisi jotenkin pystyä korjaamaan mahdollisimman hyvin. Kuitenkin tällaisten päällekkäisyyksien korjaaminen automatisoinnilla voi aiheuttaa mallin väärentymiä kyseisessä kohdassa, tai kenties jopa uusia virheitä aivan eri paikassa koko mallia.



Kuva 35. Quadri ongelmakohta päällekkäisyyksissä

Kuvan 35. tilanteessa täytyisi jotenkin saada kahden eri väylämallin luiskat liitettyä yhteen, ja ilmaan jäävä osa poistettua väylämallien penkasta. Myös luiskien alareunat ja oja täytyisi sulautua kauniisti yhteen, niin että lopputuloksen katsoja ei huomaisi mitään virhettä.

8 Mahdollisen järjestelmän käytettävyyssarvio Honeycomb menetelmällä

Suunniteltavasta järjestelmästä voidaan tehdä Morvillen Honeycomb-metodilla arviointi. Kuten kappaleessa 2.7 mainittiin, Honeycombin eri osa-alueet ovat hyödyllisyys, käyttökelpoisuus, haluttavuus, löydettävyys, saavutettavuus, uskottavuus ja arvokkus. Tässä kappaleessa tutkitaan mahdollista järjestelmää Usabilityn (2021) Honeycomb-kuvauksien avulla. Tämä arviointi tapahtuu täysin haastattelujen, omien pohdintojen sekä kahvipöytäkeskustelun perusteella.

8.1 Hyödyllisyys

Hyödyllisyyden pohtiminen voidaan aloittaa miettimällä mikä olisi koko järjestelmän tarkoitus. Koko järjestelmän tarkoitushan olisi automatisoida eri käyttötarkoituksiin muodostettavien mallien valmistusta. Samalla järjestelmän tarkoitus olisi myös vähentää visualisointiryhmän työtaakkaa automatisoimalla erilaisia työtehtäviä koko projektin elämänsäkaaren aikana.

Järjestelmä olisi hyödyllinen, jos sillä pystytään täyttämään tarve mihin se on suunniteltu ja se on monikäyttöinen muissa projekteissa tai tarpeissa. Tämän tutkimuksen perusteella järjestelmällä voitaisiin täyttää edellä mainitut kaksi tarvetta, sekä mahdollisesti muitakin tarpeita, joita tässä tutkimuksessa ei ole ilmennyt. Toisinsanottuna järjestelmä olisi hyödyllinen.

8.2 Käyttökelpoisuus

Järjestelmän käyttökelpoisuutta on suunnitteluvaiheessa hieman vaikeata arvioida. Kuitenkin selkeällä käyttöliittymällä ja ennakoon tutkituilla mahdollisilla ongelmilla voidaan lisätä käyttökelpoisuuden todennäköisyyttä.

8.3 Haluttava

Haluttavuus erilaisten kuvien-, sekä muotoiluelementtien kautta ei toteutuisi tämän järjestelmän suunnitelman perusteella. Tähän on kuitenkin tekninen syy, joka on käsitelty käyttöliittymän kappaleessa. Suuret tietomallit vievät paljon laskentatehoa, joten käyttöliittymä kannattaa pitää mahdollisimman yksinkertaisena. Toisaalta kun malli on saatu laskettua, sen näkeminen voi vaikuttaa käyttäjän tunteisiin ja herättää arvostusta käyttäjässä.

8.4 Löydettävä

Järjestelmän käyttöliittymäesimerkki on valmistettu löydettävyyttä ajatellen. Käyttöliittymä etenee loogisesti ylhäältä alaspäin, ja alavetovalikoiden nimet ovat kutakin ryhmää kuvaavia. Näin siis järjestelmä olisi esimerkkinä toimivan käyttöliittymän perusteella löydettävä.

Ongelmaksi löydettävyyden kannalta voi ilmetä itse käyttäjä. Väylämallintamisesta ja muista alaan liittyvistä asioista tietämättömällä voi olla vaikeuksia löytää ja muodostaa juuri hänen haluamansa malli. Toisaalta pienellä käyttöliittymän tutkimisella pitäisi käyttäjän päästä tämän ongelman yli.

8.5 Saavutettava

Järjestelmä on saavutettavissa, kun kaiken tasoiset käyttäjät voivat ja osaavat sitä käyttää. Saavutettavuuteen liittyvät niin värisokeat, kuin 11-vuotiaat lapset. Kaikkien pitäisi pystyä käyttämään järjestelmää. Järjestelmän käyttöympäristön kannalta käyttäjät tuskin tulevat olemaan nuorempia lapsia, joten heidät voidaan poistaa laskuista tässä

pohdinnassa. Jos järjestelmä tullaan rakentamaan, saavutettavuuden mahdollisuuksia täytyy tarkastella tarkemmin.

8.6 Uskottava

Uskottavuutta pystytään parhaiten osoittamaan vasta, jos järjestelmä rakennetaan ja pystytään tuottamaan sen avulla malleja. Suunnitelman uskottavuutta voidaan lisätä tutkimalla tämän tutkimuksen eri lähteitä ja asioita mitä tässä tutkimuksessa on otettu huomioon.

8.7 Arvokas

Arvokkuuden mittaukseen vaikuttavat suuresti käyttäjät. Järjestelmän suurin arvo tulisi visualisointiryhmälle. Hyötyjen arvioiminen olisi mahdollista laatia tarkemmin, jos järjestelmä olisi käytössä ja siitä olisi käyttökokemusta. Työmäärien arviointiin ja niiden automatisointiin tulisi laatia erillinen seurantaprojekti.

9 Mahdolliset käyttötapaukset

Järjestelmän mahdollisia käyttötapauksia voidaan pohtia tutkimalla Rambollin esimerkki projektiaikataulua kuvassa 36. Aikataulu sisältää useamman kohdan, josta voisi olla hyödyllistä tehdä visuaalinen malli. Mitä useampi malliversio samasta projektista luodaan, sitä enemmän projektissa on näytettävää muille projektiin liittyville ihmisille kuten alueen asukkaille tai esimerkiksi medialle. Näitä malleja voidaan myös käyttää hyväksi uudelleen erilaisissa suunnittelutehtävissä.

TIESUUNNITELMAN OSA-ALUEIDEN SUUNNITTELU
Suunnitteluperusteet
TS10 kustannussuunnittelu ja kustannusten arviointi
TS30 lähtötiedot, lähtötietomalli
TS40 Liikenne-ennuste ja verkolliset tarkastelut
TS40 tien linjauksen ja korkeusaseman suunnittelu
TS50 tien poikkileikkauksen ja rakenteiden suunnittelu
TS60 liittymien ja muiden järjestelyiden suunnittelu
TS70 geotekninen suunnittelu
TS80 kuivatuksen suunnittelu
TS90 ympäristösuunnittelu
TS90 meluselvitys
TS100 valaistuksen suunnittelu
TS110 liikenteenohjaus suunnittelu
TS120 tienpitäjän varusteet ja laitteet sekä johdot
TS130 johtosiirrot
TS140 läjitysalueet
TS150 siltojen ja taitorakenteiden suunnittelu
TS160 vaikutusten arviointi
TS170 lunastettavien alueiden ja oikeuksien määrittäminen
TS180 tieverkon hallinnollisten muutosten määrittely
TS190/RS220 Asiakirjojen kokoaminen
TS200 hyväksymisehdotuksen laatiminen
Tietomalli

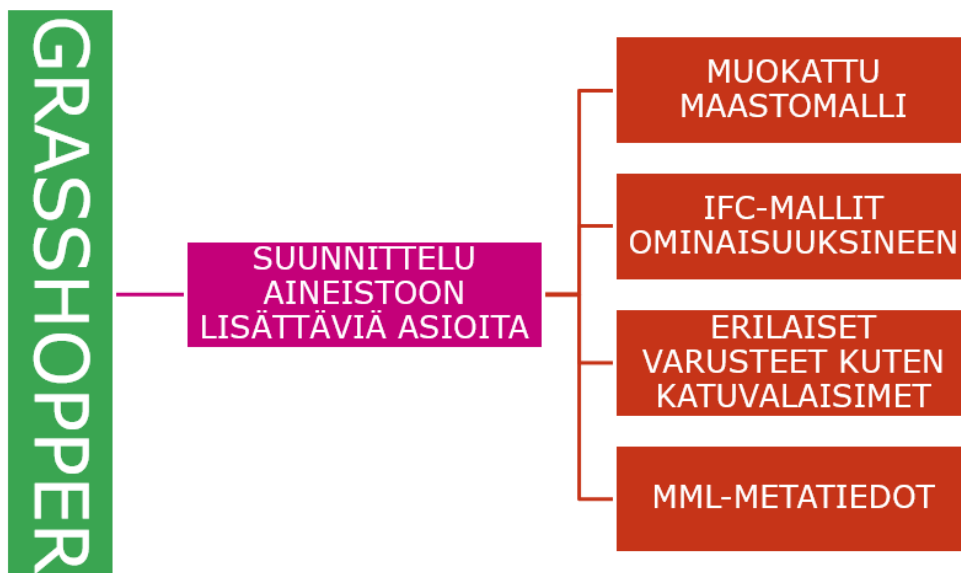
Kuva 36. Mahdollisia mallinnuspisteitä

Tällaisia kohtia projektiaikataulussa ovat esimerkiksi lähtötiedot eli niin sanottu ”lähtötietomalli”. Ajallisesti seuraava malliversio voisi olla ”liittymien ja muiden järjestelyiden

suunnittelu”. Valaistusta ja liikenteenohjausta voitaisiin myös mahdollisuuksien mukaan mallintaa, erityisesti valaistuksesta voitaisiin valmistaa erittäin elokuvamaisia 3D-malleja pelimoottorin avulla. Viimeisenä tulisi viimeistely ja täysin valmis 3D-malli projektista.

Kuten kappaleen alussa mainittiin, useammassa mallissa on omat hyötynsä. Ensimmäisessä mallissa voidaan huomata asioita mitä ei välttämättä ole tullut suunnitelmissa vastaan, ja näin mahdolliset virheet sekä erehdykset voidaan korjata heti projektin alkuvaiheissa. Alueen asukkaat voivat myös kertoa omia näkemyksiään jo projektin alkuvaiheissa ja näin varmistetaan, että mahdollisimman montaa henkeä on kuultu.

Järjestelmällä luotuja malleja voitaisiin myös ottaa järjestelmän kiertokulkuun uudelleen mukaan (kuva 37.). Luotu malli voitaisiin viedä takaisin suunnitteluvaiheeseen ja näin jälleen muokata sitä sekä käyttää hyväksi sen sisältämiä tietoja. Näin toimimalla järjestelmälle saataisiin muitakin käyttötarpeita kuin vain visualisointiin tarkoitettuja malleja.



Kuva 37. Järjestelmän kautta saatavia tietoja takaisin suunnitteluun

10 Yhteenveto

Onnistunut visualisoinnin automatisointijärjestelmä, joka toimisi kaikkien tietokantojen kanssa on hyvin vaikeaa tehdä. Oikeanlaisella suunnittelulla voidaan mahdollistaa järjestelmän tehokas rakentaminen, sillä tässä tutkimuksessa kirjatut ominaisuudet ja haasteet tulisivat erittäin todennäköisesti vastaan järjestelmän rakentamisen yhteydessä. UML kielen käyttö tarjoaa suuren avun järjestelmän suunnittelussa, siitä on helppo tulkitä mitä järjestelmän tulisi tehdä, mitä asioita täytyy ottaa huomioon, mutta eritoten muiden ammattilaisten tarjoamat kommentit täytyy huomioida, jos järjestelmää ryhdyttäisiin rakentamaan.

Tietojärjestelmätutkimusmalli on helpottanut tutkimusta tarjoamalla tutkimukselle selkeämmän rungon, ja koska käytössä oli suunnittelu ja kehityskeskineen lähestymistapa, pystyttiin keskittyä alusta alkaen järjestelmän suunnitteluun. Suunnittelussa tullaan vielä tekemään tarkempia UML kaavioita alan ammattilaisten suorittaman arvioinnin jälkeen. Tietojärjestelmätutkimusmallia tullaan myös käyttämään tulevaisuudessa, jos järjestelmä tullaan rakentamaan, sillä osia malliin on jo syntynyt tässä tutkimuksessa.

Maanmittauslaitokselta saatavaa tietoa on valtava määrä niin teiden nimistä puuston korkeuteen ja tämän kaltaista tietoa voitaisiin käyttää visualisoinnin lisänä. Avoimia rajapintoja on muitakin kuin maanmittauslaitos ja myös niissä voisi olla hyödyllistä tietoa visualisoinnin parantamiseen. Olisi ehkä mahdollista lukea palvelimelta rakennusten seinien väriä, kattojen materiaalia sekä ikkunoiden sijaintia rakennuksesta. Kaikki tämän kaltainen tieto lisäisi visualisoinnin laatua katsojan silmissä ja kaikki tieto mikä saadaan automatisoinnilla vähentää seuraavan työryhmän työmäärää. Näyttämällä todenmukainen malli esittelytilaisuuksissa, välttyttäisiin epäolennaisilta kysymyksiltä, jotka koskisivat malliin tulleita vääriä materiaaleja tai tietoja.

Visualisoitavia kohteita miettiessä on kuitenkin syytä muistaa, että kaikki tieto lisää taas järjestelmän raskaita toimenpiteitä ja itse mallin tiedostokokoa. Tehokkailla ns. pelikoneilla kaikenlaiset mallit toimivat varmasti hyvin, mutta jos järjestelmän käyttäjällä on

käytössä aivan perinteinen tietokone, voi olla, että tietokoneen tehokkuus ei riitä järjestelmän ja mallin hyödyntämiseen.

Tietokoneiden rajallisuudesta pääsemme seuraavaan mahdolliseen kehityskohteeseen; järjestelmän siirtämisen tehokkaaseen virtuaalitietokoneeseen. Virtuaalitietokone mahdollistaisi sen, että järjestelmän käyttäjä käskisi virtuaalitietokonetta suorittamaan mallin luomisen samalla kun itse tekisi muita työasioita omalla koneellaan. Kenties tällainen virtuaalitietokone tarjoaisi oman käyttöliittymän internetselaimen välityksellä ja antaisi käyttäjälle sitten lopputuotoksena valmiin mallin haluttujen kriteerien perusteella.

Lähteet

3DNatives. (2020). *Rhino: what are the features of this 3D modeling software?*. 3Dnatives. Noudettu 2020-12-10 osoitteesta <https://www.3dnatives.com/en/rhino-3d-modeling-software-080420205/>

Alfame. (2016). *Järjestelmäintegraatio tehostaa toimintaa - 3 integraation hyötyä liiketoiminnalle*. Alfame. Noudettu 2020-12-01 osoitteesta <https://www.alfame.com/blog/jarjestelmaintegraatio-tehostaa-toimintaa-3-integraation-hyotya-liiketoiminnalle>

Avoin rajapinta. (2014, 11. lokakuuta). *Avoimen rajapinnan määritelmä*. Avoin rajapinta. Noudettu 2021-04-15 osoitteesta <http://avoinrajapinta.fi/>

Baldwin, M. (2018). *What is openBIM?*. BIMconnect. Noudettu 2021-04-21 osoitteesta <https://bimconnect.org/en/wiki/what-is-openbim/>

Baldwin, M. (2018). *What is IFC?*. BIMconnect. Noudettu 2021-04-21 osoitteesta <https://bimconnect.org/en/software/what-is-ifc/>

BuildingSMART. (2021). *OpenBIM*. BuildingSMART. Noudettu 2021-04-21 osoitteesta <https://www.buildingsmart.org/about/openbim/>

Compean, S. (201). *Selecting an Enterprise Operating Model Based on the Business Model Design*. SogetiLabs. Noudettu 2021-04-22 osoitteesta <https://labs.sogeti.com/selecting-an-enterprise-operating-model-based-on-the-business-model-design/>

Ekpo, S. & George, D. (2010-5-8. huhtikuuta). A system-based design methodology and architecture for highly adaptive small satellites. *2010 IEEE International Systems Conference*. 516-519. IEEE. doi: 10.1109/SYSTEMS.2010.5482323.

Epic Games. (2021). *Unreal Engine Features*. Epic Games. Noudettu 2021-4-3 osoitteesta <https://www.unrealengine.com/en-US/features>

FILEFORMAT. (2021). *What is an IFC file?* FILEFORMAT. Noudettu 2021-04-21 osoitteesta <https://docs.fileformat.com/cad/ifc/>

GeeksforGeeks. (2019). *Unified Modeling Language (UML) | An Introduction*. GeeksforGeeks. Noudettu 2021-04-12 osoitteesta <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>

Graafinen. (2015). *Hyvä käyttöliittymä - 10 muistisääntöä*. Graafinen. Noudettu 2021-04-14 osoitteesta <https://www.graafinen.com/suunnittelu/digi/hyva-kayttoliittyma-10-muistisaantoa/>

Hevner, A., March, S., Park, J. (2004, March). *Design Science in Information Systems Research*. ResearchGate. Noudettu 2021-03-22 osoitteesta https://www.researchgate.net/publication/201168946_Design_Science_in_Information_Systems_Research

Hevner, A. & Gregor, S. (2013, Kesäkuu). *Positioning and Presenting Design Science Research for Maximum Impact*. ResearchGate. Noudettu 2021-03-22 osoitteesta https://www.researchgate.net/publication/262350911_Positioning_and_Presenting_Design_Science_Research_for_Maximum_Impact

Hoogenraad, W. (2018, Lokakuu). *Sovellus, järjestelmä, työkalu tai ohjelmisto, mitä se on?* ITpedia. Noudettu 2021-04-10 osoitteesta <https://fi.itpedia.nl/2018/10/23/applie-systeem-tool-of-software-wat-is-wat/>

- Hsu, T., Tsai, M., Babu, S., Hsu, P., Chang, H., Lin, W., Chuang, J. (2020-22-26. maaliskuuta). Design and Initial Evaluation of a VR based Immersive and Interactive Architectural Design Discussion System. *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR.)* 363-371. IEEE. doi: 10.1109/VR46266.2020.00056.
- Jaiwai, M. & Sammapun, U. (2017, 12-14. heinäkuuta). Extracting UML class diagrams from software requirements in Thai using NLP. *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. 1-5. IEEE. doi: 10.1109/JCSSE.2017.8025938.
- Kilkelly, M. (2016-15. huhtikuuta). *5 Ways Computational Design Will Change the Way You work*. ArchDaily. Noudettu 2021-04-21 osoitteesta <https://www.archdaily.com/785602/5-ways-computational-design-will-change-the-way-you-work>
- Li, Y., Li, M. & Liu, P. (2013, 20-22. kesäkuuta). Research on a coal mine virtual system using Unreal Engine. *2013 21st International Conference on Geoinformatics*. 1-5. IEEE. doi: 10.1109/Geoinformatics.2013.6626154.
- Lo, C. & Huang, S. (2010-18-21. elokuuta). Embedded control system development using UML for automatic doors. *Proceedings of SICE Annual Conference 2010*. 3332-3335. Noudettu 2021-04-16 osoitteesta <https://ieeexplore-ieee-org.proxy.uwasa.fi/document/5602536> Ei DOI numeroa.
- Longbotham, N., Kontgis, C. & Maguire, C. (2018-22-27. heinäkuuta). Harmonization And Fusion of Global Scale Data. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. 1738-1739. IEEE. doi: 10.1109/IGARSS.2018.8519041.

- Macpherson, E. (2019-7. lokakuuta). *The UX Honeycomb: Seven Essential Considerations for Developers*. Medium. Noudettu 2021-04-16 osoitteesta <https://medium.com/mytake/the-ux-honeycomb-seven-essential-considerations-for-developers-accc372a398c>
- Martin, G. (2003-20. kesäkuuta). The reality of system design today: do theory and practice meet? *Third International Conference on Application of Concurrency to System Design*. IEEE. doi: 10.1109/CSD.2003.1207692.
- Morville, P. (2004, 21. kesäkuuta). *User Experience Design*. Semantic Studios. Noudettu 2021-04-14 osoitteesta http://semanticstudios.com/user_experience_design/
- Muranen, A & Harmainen, L. (2021). *Käyttöliittymä- & käyttäjäkokemussuunnittelu (UI & UX Design)*. itewiki. Noudettu 2021-04-14 osoitteesta <https://www.itewiki.fi/opas/kayttoliittymasuunnittelu-ux-user-experience-design-eli-kayttajakokemus/>
- Paikkatietojen rajapintapalvelut. (2021). *Paikkatietojen rajapintapalvelut*. Maanmittauslaitos. Noudettu 2021-04-21 osoitteesta <https://www.maanmittauslaitos.fi/rajapinnat/paikkatiedot>
- Pefers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W, Virtanen, V., & Bragge, J. (2006, February). *The design science research process: A model for producing and presenting information systems research*. ResearchGate. Noudettu 2020-12-02 osoitteesta https://www.researchgate.net/publication/228650671_The_design_science_research_process_A_model_for_producing_and_presenting_information_systems_research

Quadri. (2021). *Quadri Integrations*. Trimble. Noudettu 2021-4-3 osoitteesta <https://quadri.trimble.com/integrations>

RAKLI. (2021). *RAKLI edistää tiedon harmonisointia*. RAKLI ry. Noudettu 2021-04-21 osoitteesta <https://www.rakli.fi/digitalisaation-edistaminen/tiedon-harmonisointi/>

Ramboll. (2021). *Yritys*. Ramboll. Noudettu 2021-3-15 osoitteesta https://fi.ramboll.com/ramboll_finland_oy

Ramboll. (2021). *Rakennetun ympäristön digiratkaisut*. Ramboll. Noudettu 2021-3-15 osoitteesta https://fi.ramboll.com/ramboll_finland_oy/innovaatiot-ja-digitaaliset-ratkaisut

Ren, X., Sun, J., Xing, Z., Xia, X., Sin & Sun, J. (2020, 5-11. lokakuuta) Demystify Official API Usage Directives with Crowdsourced API Misuse Scenarios, Erroneous Code Examples and Patches. *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. 925-936. IEEE. Ei doi numeroa.

Reubens, R. (2016). *To Craft, by design, for sustainability Towards holistic sustainability: design for developing-country enterprises*. Delft University of Technology. Noudettu 2020-12-02 osoitteesta <http://resolver.tudelft.nl/uuid:0c2c14c8-9550-449d-b1ff-7e0588ccd6c2>

Rhinoceros. (2020). *Rhinoceros Features*. Rhinoceros. Noudettu 2020-12-10 osoitteesta <https://www.rhino3d.com/features/>

Röyskö, E. (2021, 15. huhtikuuta). *Tiedonhallinnan rooli eri organisaatioiden menestykselle kasvaa*. WellBit. Noudettu 2021-04-21 osoitteesta <https://www.wellbit.fi/blogi/tiedonhallinnan-rooli-eri-organisaatioiden-menestykselle-kasvaa/>

SmartDraw. (2021). *Activity Diagram*. SmartDraw. Noudettu 2021-04-12 osoitteesta <https://www.smartdraw.com/activity-diagram/>

Syrjä, M. (2018-26. helmikuuta). *LandXML-tiedostojen luku*. 3D-Win Wiki. Noudettu 2021-04-21 osoitteesta <http://www.3d-system.net/wiki/index.php/tiedosto/formaatit/20-landxml-tiedostojen-luku>

Trimble. (2021). *Quadri Capabilities*. Trimble. Noudettu 2021-4-3 osoitteesta <https://quadri.trimble.com/capabilities>

Tutorialspoint. (2021). *UML - Class Diagram*. Tutorialspoint. Noudettu 2021-4-12 osoitteesta https://www.tutorialspoint.com/uml/uml_class_diagram.htm

Usability. (2021). *User Experience Basics*. Usability. Noudettu 2021-4-14 osoitteesta <https://www.usability.gov/what-and-why/user-experience.html>

Valjas. (2021). *Mitä integraatio, rajapinta ja api tarkoittavat?* Valjas. Noudettu 2021-04-14 osoitteesta <https://valjas.fi/mita-integraatio-rajapinta-ja-api-tarkoittavat/>

Visual Paradigm. (2020). *UML Class Diagram Tutorial*. Visual Paradigm. Noudettu 2021-4-12 osoitteesta <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

Visual Paradigm. (2020). *What is Unified Modeling Language (UML)?* Visual Paradigm. Noudettu 2021-4-12 osoitteesta <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>

Wang, Q. Yang, A. & Ren, C. (2011). *A Method of 2D Digital Mankind Model Reconstruction Based on Rhinoceros*. Noudettu 2021-04-15 osoitteesta <https://ieeexplore-ieee-org.proxy.uwasa.fi/document/5997635>

Wilson, J. (2002). *Responsible Authorship and Peer Review (8:2)*. Science and Engineering Ethics (s.155-174)

Xamk. (2021). *DATAVISUALISOINTIOPAS-VISUALISOINTI*. Xamk. Noudettu 2021-04-15 osoitteesta <https://www.xamk.fi/dataopas-visualisointi/>